자율프로젝트 금융팀

Dacon 제주 신용카드 빅데이터 경진대회

공모전 소개

Dacon 제주 신용카드 빅데이터 경진대회

주제 | AI 알고리즘 활용 카드 사용 금액 예측

목표 | **주어진 2019년 1월 – 2020년 4월 카드 사용 내역 데이터**를 활용해
2020년 7월의 지역별, 업종별 카드 사용 총액 예측

배경 | 신용카드 사용량을 분석을 통한 'Post COVID-19 시대' 신용카드 사용량 예측 모델 개발
지역 경제 위축 및 중소상공인 경영난 해소

공모전 소개

Dacon 제주 신용카드 빅데이터 경진대회

기대효과 | 해당 모델을 통해 카드 사용금액에 미치는 코로나의 영향을 잘 학습한다면 장기전으로 예상되는 코로나 시대에 있어서 여러 경제상황을 예측하는 데에 도움이 될 수 있을 것으로 기대됨.

데이터 : 2019.01~2020.04 카드 사용 내역 데이터

Dacon 제주 신용카드 빅데이터 경진대회

데이터셋 구성

X

결제일자 (년월)	가구생애주기
카드이용지역 (시도)	이용고객수
카드이용지역 (시군구)	이용 건수
업종명	연령대
거주지역 (시도)	성별
거주지역 (시군구)	

Y

이용금액

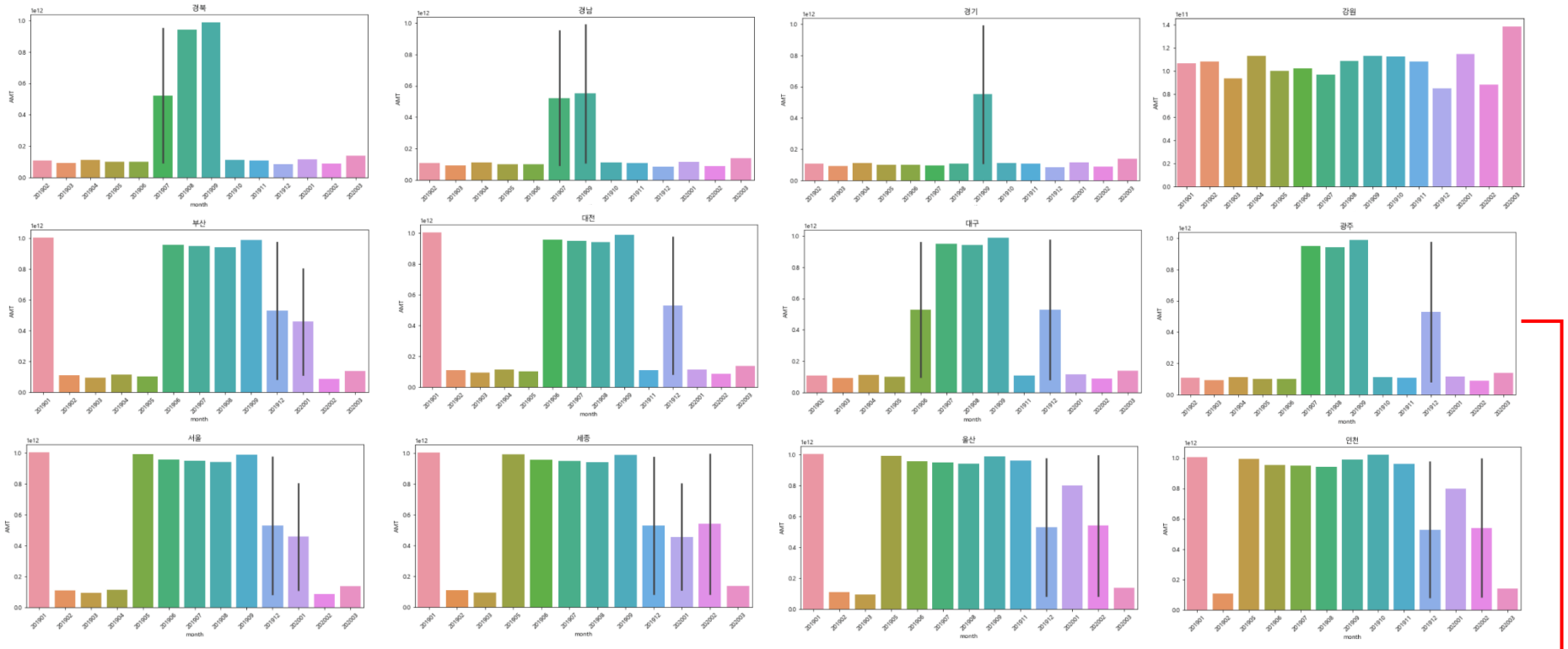
데이터 : 2019.01~2020.04 카드 사용 내역 데이터

Dacon 제주 신용카드 빅데이터 경진대회

REG_YYMM	CARD_SIDO_NM	CARD_CCG_NM	STD_CLSS_NM	HOM_SIDO_NM	HOM_CCG_NM	AGE	SEX_CTGO_CD	FLC	CSTMR_CNT	AMT	CNT
201901	강원	강릉시	건강보조식품 소매업	강원	강릉시	20s	1	1	4	311200	4
201901	강원	강릉시	건강보조식품 소매업	강원	강릉시	30s	1	2	7	1374500	8
201901	강원	강릉시	건강보조식품 소매업	강원	강릉시	30s	2	2	6	818700	6
201901	강원	강릉시	건강보조식품 소매업	강원	강릉시	40s	1	3	4	1717000	5
201901	강원	강릉시	건강보조식품 소매업	강원	강릉시	40s	1	4	3	1047300	3
...	...	...	...	...	...	...	...	...	...	...	...
202004	충북	충주시	휴양콘도 운영업	충북	충주시	20s	1	1	5	77000	5
202004	충북	충주시	휴양콘도 운영업	충북	충주시	30s	1	2	6	92000	6
202004	충북	충주시	휴양콘도 운영업	충북	충주시	40s	2	3	5	193000	5
202004	충북	충주시	휴양콘도 운영업	충북	충주시	50s	1	4	5	86000	7
202004	충북	충주시	휴양콘도 운영업	충북	충주시	60s	2	5	3	227000	4

EDA : 시도별 월별 이용금액

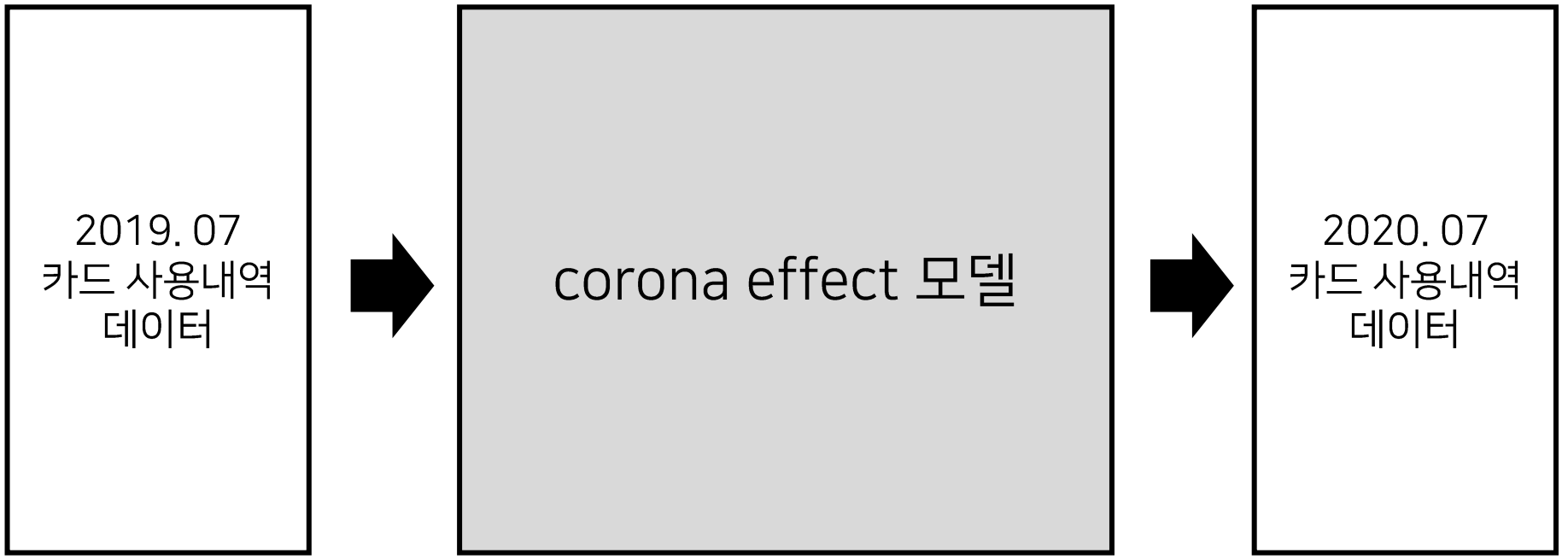
Dacon 제주 신용카드 빅데이터 경진대회



월별 카드 사용금액 차이가 큼

Model Insight

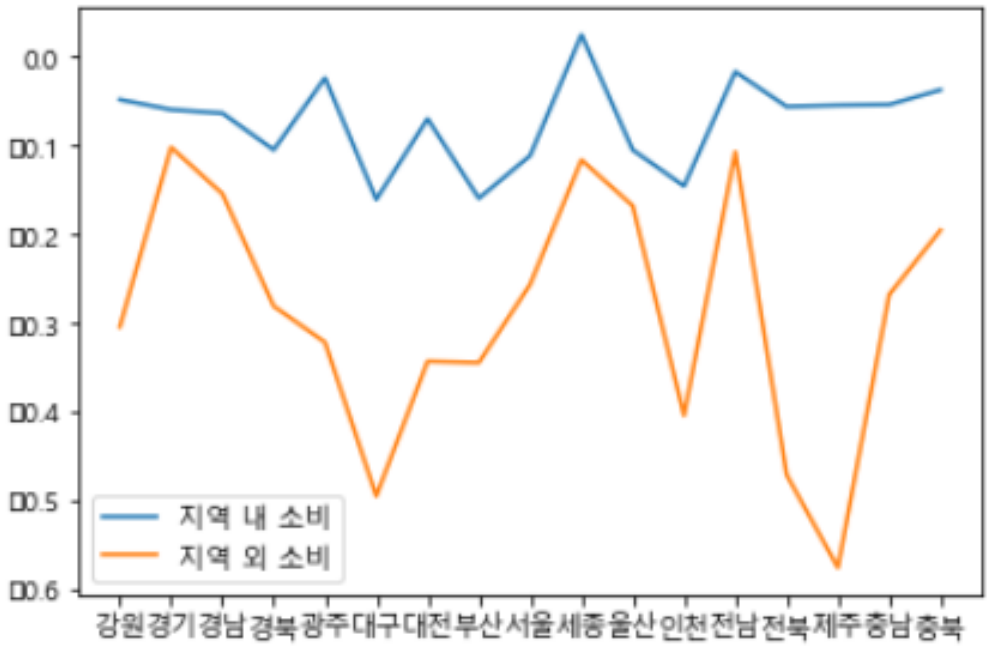
Dacon 제주 신용카드 빅데이터 경진대회



EDA – "지역 내 소비"와 "지역 외 소비" 코로나 전후 카드 사용액 증감율

Dacon 제주 신용카드 빅데이터 경진대회

- 지역 내 소비와 지역 외 소비를 분리하여 예측하기 위해 home / out으로 파일 분할
  : 지역 내 소비와 지역 외 소비 각각에 대한 코로나 영향력에 차이가 있으므로 분리하여 예측



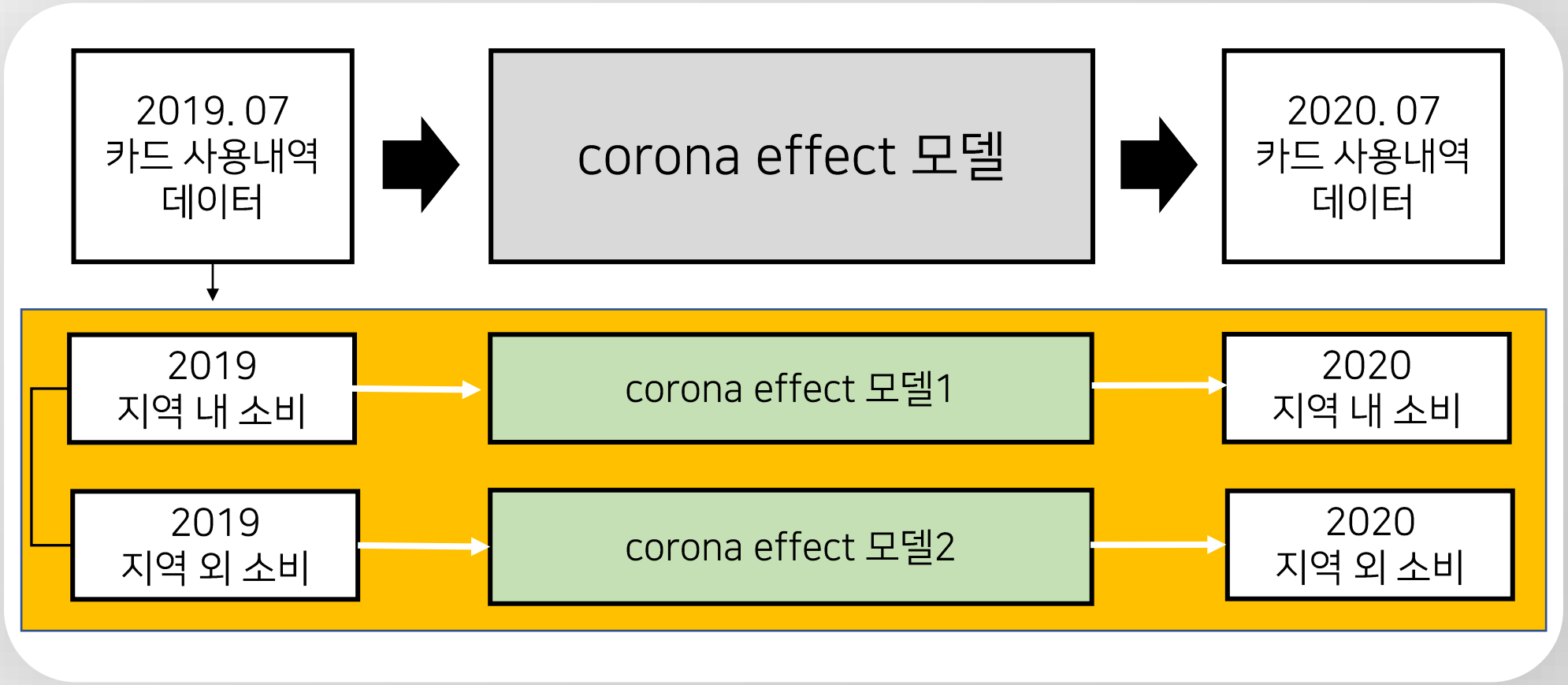
지역 내 소비 감소율

지역 외 소비 감소율

** 지역 내 소비: 거주 지역 내에서 발생하는 소비*
** 지역 외 소비: 거주 지역 외에서 발생하는 소비*

Model Insight

Dacon 제주 신용카드 빅데이터 경진대회



EDA – 관광지 설정

Dacon 제주 신용카드 빅데이터 경진대회

관광지 : 지역 외 소비 지역 비율이 높음 + 지역 외 소비가 30% 이상 감소한 지역

- 관광지 설정

places	me_difference	out_difference
제주	-0.055061	-0.575305
대구	-0.160988	-0.494736
전북	-0.056514	-0.470712
인천	-0.145881	-0.404053
부산	-0.159715	-0.344622
대전	-0.070147	-0.343134
광주	-0.024200	-0.321704
강원	-0.048690	-0.304372
경북	-0.105129	-0.281015
충남	-0.054341	-0.268043
서울	-0.111579	-0.256540
충북	-0.037603	-0.195590
울산	-0.105217	-0.168110
경남	-0.063896	-0.154563
세종	0.024371	-0.116550
전남	-0.017273	-0.107443
경기	-0.059986	-0.102287

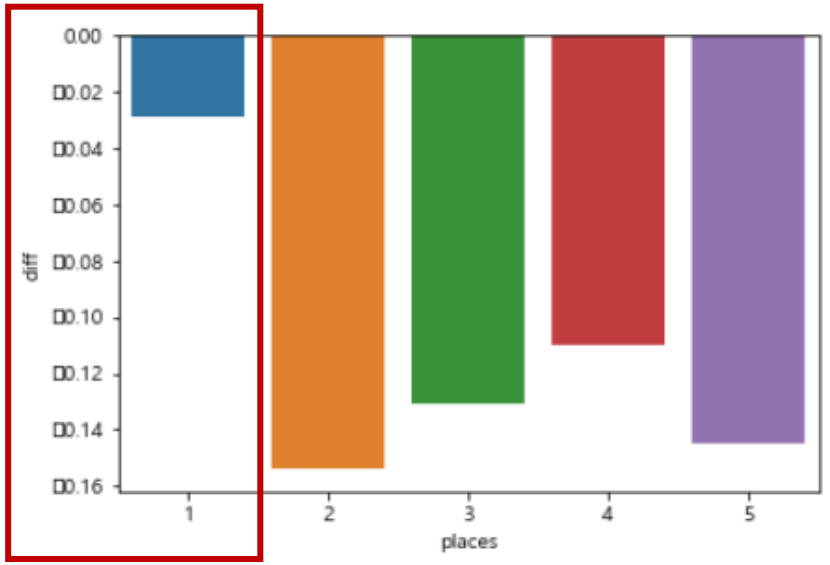
&

CARD_SIDO_NM	home	out	out/total
서울	10092021194371	4585109854660	0.312398
제주	1017397760074	326184464694	0.242772
강원	1204443934968	385167364951	0.242303
세종	156965899441	47079917907	0.230732
인천	2374676626919	681570835994	0.223009
경북	2234756004003	532864591245	0.192535
충남	1719294865555	383902509553	0.182533
전남	1556688799700	283071977439	0.153863
충북	1245478356168	219958829755	0.150098
부산	3821544401090	669761630451	0.149124
경기	12202350438264	1999534796605	0.140794
대전	986590235182	142323225534	0.126071
경남	3108803198778	376326831086	0.107981
울산	1138595760471	116617845269	0.092907
전북	1797022483446	164207701895	0.083727
광주	1582501232195	123149128739	0.072201
대구	2810736165542	209328187062	0.069312

EDA – 가구 크기별 코로나 전후 카드 사용액 증감율

Dacon 제주 신용카드 빅데이터 경진대회

- 1인 가구의 특성을 고려하기 위해 1인 가구 / 2인 이상 가구로 범주화



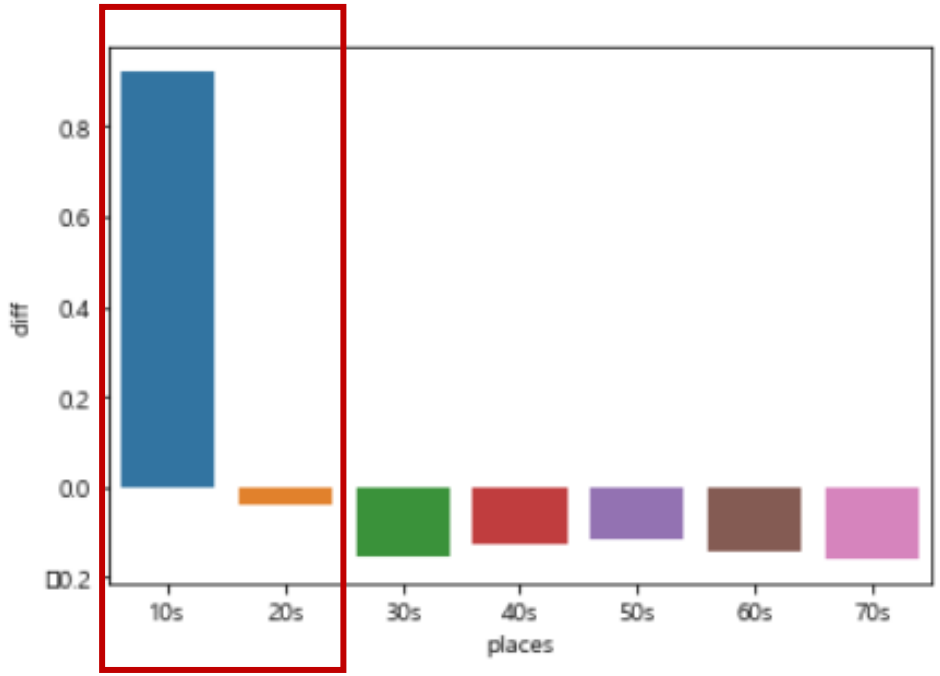
	places	before_amt	after_amt	diff
0	1	902939113648	876745851441	-0.029009
1	2	1654546944545	1399411874007	-0.154202
2	3	1499365144115	1302931384584	-0.131011
3	4	2297502982092	2044640018723	-0.110060
4	5	1385401407487	1184850242878	-0.144760

1인가구 -> 다른 가구에 비해 카드 사용액 감소폭 적음

EDA – 나이별 코로나 전후 카드 사용액 증감율

Dacon 제주 신용카드 빅데이터 경진대회

- 1020대/다른 연령대를 구분하여 0과 1로 범주화



	places	before_amt	after_amt	diff
0	10s	13647354819	26216217907	0.920974
1	20s	859504583457	825365248011	-0.039720
2	30s	1510519326405	1275246287504	-0.155756
3	40s	1934676175608	1692836457609	-0.125003
4	50s	2036006744111	1804064917724	-0.113920
5	60s	1116830903992	958871155281	-0.141436
6	70s	268570503495	225979087597	-0.158586

1020대 -> 다른 연령에 비해 카드 사용액 감소폭 적음

EDA – 업종별 코로나 영향

Dacon 제주 신용카드 빅데이터 경진대회

- 업종별 코로나 영향

places	before_amt	after_amt	difference
여행사업	16025918056	1446851167	-0.909718
정기 항공 운송업	87667346835	13218273648	-0.849222
면세점	59506546481	16359142725	-0.725087
그외 기타 스포츠시설 운영업	493784310	149674740	-0.696882
전시 및 행사 대행업	11708282695	3990706772	-0.659155
휴양콘도 운영업	9483029516	4411560753	-0.534794
버스 운송업	31752768164	15518011184	-0.511286
내항 여객 운송업	5158577751	2557760302	-0.504173
과실 및 채소 소매업	79632440208	77199614502	-0.030551
차량용 주유소 운영업	919474014250	905930910376	-0.014729
골프장 운영업	58281788156	57627693076	-0.011223
체인화 편의점	626779589018	658189792989	0.050114
그외 기타 종합 소매업	128565957911	135034988539	0.050317
육류 소매업	149456616745	157574816281	0.054318
기타음식료품위주종합소매업	195291938261	214772649001	0.099752
슈퍼마켓	841080677295	977004137225	0.161606
자동차 임대업	2604172739	3881401536	0.490455

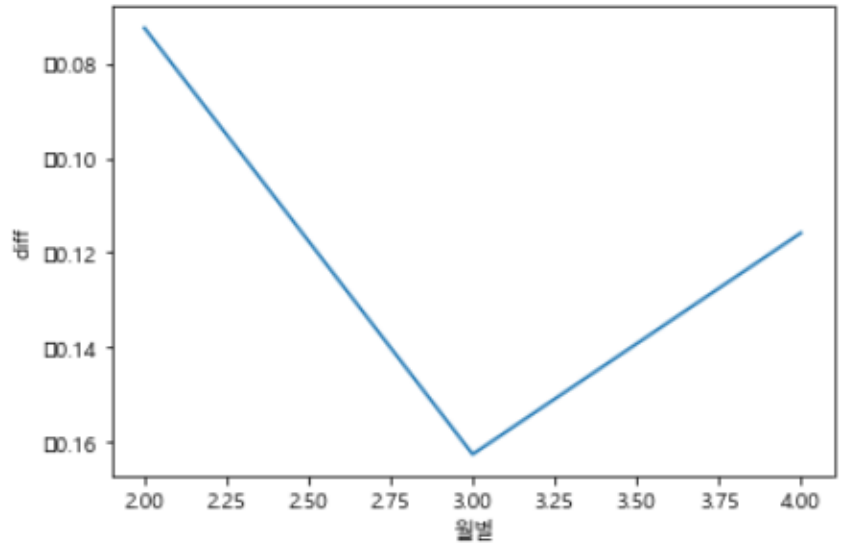
여행관련 사업 증감율 : -91%
자동차 임대업 : +5%

→ 업종별 corona effect 차이가 큼

EDA – 월별 코로나 영향

Dacon 제주 신용카드 빅데이터 경진대회

- 월별 코로나 영향 확인



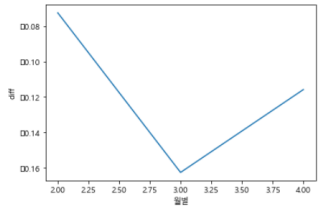
월별 코로나 영향이 다름

변수 선택 및 추가

Dacon 제주 신용카드 빅데이터 경진대회

- 월별 코로나 영향 확인

월별 코로나 영향을 반영할 수 있는 지표



코로나 검색량

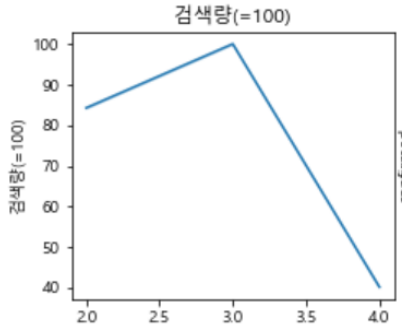
출처 : 네이버 데이터랩

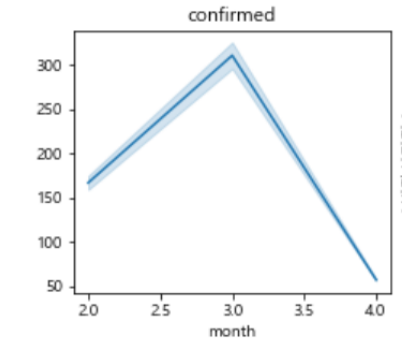
코로나 확진자수

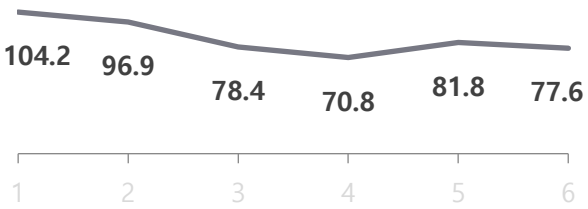
출처 : kaggle

소비자 심리지수

출처 : 국가통계포털 KOSIS







최종 feature

Dacon 제주 신용카드 빅데이터 경진대회

corona effect model

feature 명	type
결제 월	one-hot encoding
카드이용지역(시도)	one-hot encoding
업종(one-hot encoding)	one-hot encoding
연령대(1020, 그외로 0,1 labeling)	0(1020s) / 1(그 외)
가구별생애주기	0(1인가구) / 1(그 외)
코로나 확진자수	float
소비자 심리지수	float
업종별 코로나 이후 증감율	float
관광지 여부(0,1 labeling)	0(관광지X) / 1(관광지 O)

참조 - EDA 및 전처리 코드

Dacon 제주 신용카드 빅데이터 경진대회

In [7]:
```
# 필요없는 피처 제거

bm = bm.drop(['CARD_CCG_NM', 'HOM_CCG_NM', 'SEX_CTGO_CD', 'CSTMR_CNT', 'CNT'], axis=1)
```

In [9]:
```
# 1인 가구 / 나머지 구분

bm['FLC'] = bm['FLC'].apply(lambda x: 'single' if x == 1 else 'family')
```

In [8]:
```
# 10-20대 / 나머지 세대 구분

bm['AGE'] = bm['AGE'].apply(lambda x: '1020s' if x == '10s'or x== '20s' else '3060s')
```

In [42]:
```
# 월별 총 카드 이용금액
# 월별, 시/도별, 업종별 amt
amt = data.groupby(["REG_YYMM","CARD_SIDO_NM","STD_CLSS_NM"])["AMT"].sum()
amt = pd.DataFrame(amt)
# 월별 amt
month_amt = data.groupby(["REG_YYMM"])["AMT"].sum()
month_amt = pd.DataFrame(month_amt.values, index= month_amt.index)
# 시/도별 amt
place_amt = data.groupby(["CARD_SIDO_NM"])["AMT"].sum()
place_amt = pd.DataFrame(place_amt.values, index= place_amt.index)
# 업종별 amt
type_amt = data.groupby(["STD_CLSS_NM"])["AMT"].sum()
type_amt = pd.DataFrame(type_amt.values, index= type_amt.index)
```

참조 - EDA 및 전처리 코드

Dacon 제주 신용카드 빅데이터 경진대회

In [105]:
```
place_month_amt = pd.DataFrame(data.groupby(["REG_YYMM","CARD_SIDO_NM"])["AMT"].sum())
```

In [138]:
```
# unique 한 값 추출
places = data["CARD_SIDO_NM"].unique()
months = data["REG_YYMM"].unique()
```

In [139]:
```
# 월, 장소 추가
place_month_amt["place"]=list(places)*(255//len(places))
place_month_amt["month"]=list(months)*(255//len(months))

# index 초기화
place_month_amt.index = range(255)
```

In [230]:
```
# 코로나 전과 후의 거주지 소비, 외적 소비 증감율 칼럼 추가
home_out["home_difference"]=(home_out["after_home"]-home_out["before_home"])/home_out["before_home"]
home_out["out_difference"] =(home_out["after_out"]-home_out["before_out"])/home_out["before_out"]

home_out.index = home_out["places"]
home_out = home_out.drop(["places"],axis=1)
```

In [231]:
```
home_out.sort_values(by="out_difference")
```

In [11]:
```
# 관광지 구분 (1,0 표시)

travel = ['제주','대구','전북','인천','부산','대전','광주','강원']
bm['out_home']= bm['CARD_SIDO_NM'].apply(lambda x: 1 if x in travel else 0)
```

In [12]:
```
bm.to_csv('관광지 구분(home, out 분리전).csv', encoding='utf-8-sig')
```

참조 - EDA 및 전처리 코드

Dacon 제주 신용카드 빅데이터 경진대회

In [310]:
```
#201902,201903을 코로나 전으로, 202002,202003을 코로나 후로 하여 데이터 분리
before_amt = data.loc[(data["REG_YYMM"]==201902) | (data["REG_YYMM"]==201903),:]
after_amt = data.loc[(data["REG_YYMM"]==202002) | (data["REG_YYMM"]==202003),:]

# 업종별로 데이터를 나눈 후 사용 금액 합산
before_amt = before_amt.groupby(["STD_CLSS_NM"])["AMT"].sum()
after_amt = after_amt.groupby(["STD_CLSS_NM"])["AMT"].sum()
```

In [320]:
```
before_after_amt = pd.DataFrame(before_amt).merge(pd.DataFrame(after_amt), on=before_amt.index)
before_after_amt.columns = ["places","before_amt", "after_amt"]
```

In [328]:
```
before_after_amt.index=before_after_amt["places"]
before_after_amt = before_after_amt.drop(["places"], axis=1)
```

In [331]:
```
before_after_amt.columns
```

Out[331]:
```
Index(['before_amt', 'after_amt'], dtype='object')
```

In [332]:
```
# 코로나 전과 후의 업종별 소비 증감율 칼럼 추가
before_after_amt["difference"]=(before_after_amt["after_amt"]-before_after_amt["before_amt"])/before_after_amt["
before_amt"]
```

In [337]:
```
before_after_amt.to_csv("./업종별_코로나매출.csv",index=True)
```

In [306]:
```
home_out
```

MODEL 구현

모델 구현

```
#피처 다합쳐진 데이터프레임 불러오기
bm_ci_search_covid_consume = pd.read_csv('./train.csv')
bm_ci_search_covid_consume.drop(['Unnamed: 0'],axis=1,inplace=True)
bm_ci_search_covid_consume
```

	CARD_SIDO_NM	STD_CLSS_NM	HOM_SIDO_NM	AGE	year	FLC	month	AMT	out_home	diff	검색량 (=100)	confirmed	소비자심리지수
0	강원	건강보조식품 소매업	강원	1020s	2019	family	2	216200	1	-0.106782	0.02146	0.0	97.1
1	강원	건강보조식품 소매업	강원	1020s	2019	single	2	1517000	1	-0.106782	0.02146	0.0	97.1
2	강원	건강보조식품 소매업	강원	3060s	2019	family	2	144433371	1	-0.106782	0.02146	0.0	97.1
3	강원	골프장 운영업	강원	1020s	2019	family	2	1629420	1	0.034461	0.02146	0.0	97.1
4	강원	골프장 운영업	강원	1020s	2019	single	2	8254950	1	0.034461	0.02146	0.0	97.1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
265430	서울	정기 항공 운송업	충북	1020s	2020	single	4	3017380	0	-0.862732	40.21097	183.0	72.4
265431	서울	정기 항공 운송업	충북	3060s	2020	family	4	37190900	0	-0.862732	40.21097	183.0	72.4
265432	인천	정기 항공 운송업	충북	1020s	2020	single	4	54700	1	-0.862732	40.21097	29.0	72.4
265433	제주	정기 항공 운송업	충북	1020s	2020	single	4	480500	1	-0.862732	40.21097	4.0	72.4
265434	제주	정기 항공 운송업	충북	3060s	2020	family	4	3678500	1	-0.862732	40.21097	4.0	72.4

265435 rows × 13 columns

늦참

기본 IDEA

모델 학습

그리드 서치

예측

최종 결과

모델 구현

```
#피처 다합쳐진 데이터프레임 불러오기
bm_ci_search_covid_consume = pd.read_csv('./train.csv')
bm_ci_search_covid_consume.drop(['Unnamed: 0'],axis=1,inplace=True)
bm_ci_search_covid_consume
```

	CARD_SIDO_NM	STD_CLSS_NM	HOM_SIDO_NM	AGE	year	FLC	month	AMT	out_home	diff	검색량 (=100)	confirmed	소비자심리지수
0	강원	건강보조식품 소매업	강원	1020s	2019	family	2	216200	1	-0.106782	0.02146	0.0	97.1
1	강원	건강보조식품 소매업	강원	1020s	20	le	2	1517000	1	-0.106782	0.02146	0.0	97.1
2	강원	건강보조식품 소매업	강원	3060s	20	ly	2	144433371	1	-0.106782	0.02146	0.0	97.1
3	강원	골프장 운영업	강원	1020s	2		2	1629420	1	0.034461	0.02146	0.0	97.1
4	강원	골프장 운영업	강원	1020s	2019	ingle	2	8254950	1	0.034461	0.02146	0.0	97.1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
265430													72.4
265431													72.4
265432													72.4
265433	제주	정기 항공 운송업	충북	1020s	2020	single	4	480500	1	-0.862732	40.21097	4.0	72.4
265434	제주	정기 항공 운송업	충북	3060s	2020	family	4	3678500	1	-0.862732	40.21097	4.0	72.4

265435 rows × 13 columns

REG_YYMM	CARD_SIDO_NM	STD_CLSS_NM	AMT
2020년 7월	지역별	업종별	카드 사용금액

모델 구현- 기본 IDEA

월별 경향성 존재

- 월별 카드 사용금액 이용 추세가 다름.
  EX) 7월 휴가철로 인한 호텔, 여행 등 업종의 카드 사용 증가

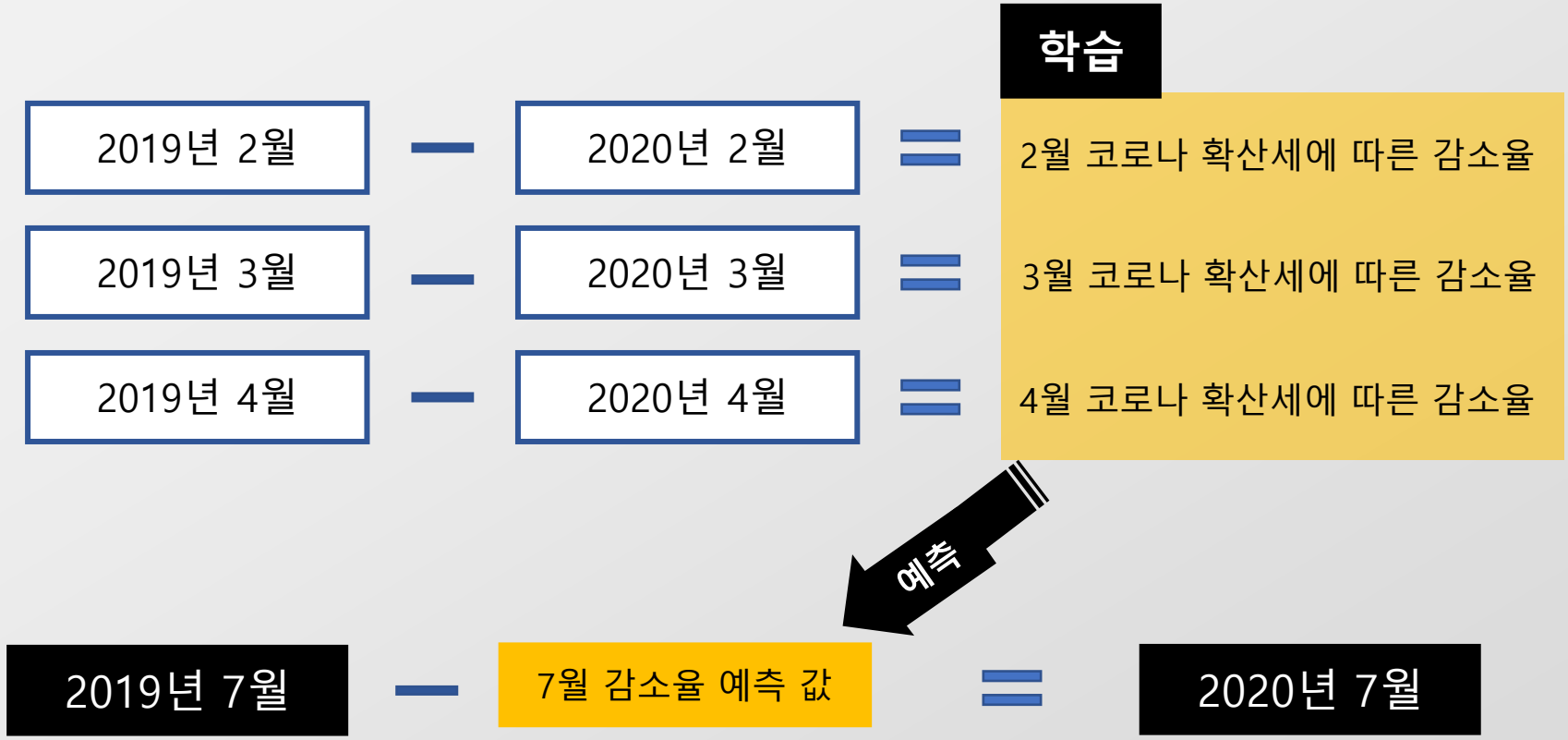
2019와 2020의 차이는 코로나

- 2020년 7월은 2019년 7월의 추세를 따라갈 것
- 그러나, 코로나의 영향에 따른 소비 감소

즉, 코로나가 없다면 전년도 월별 소비 패턴을 유사하게 따라가나
지역별, 업종별로 코로나로 인한 카드 사용 감소율이 달라질 것

=우리 모델의 TARGET

모델 구현- 기본 IDEA



거주지 내에서의 카드 사용금액 변화와 **거주지 밖에서의 카드 사용금액 변화**는
다른 양상을 보였으므로 **HOME/OUT 구분하여 학습하고 합치기.**

모델 학습

2019년 2,3,4월과 2020년 2,3,4월 차이

```
#2019년 234월 데이터 뽑아내기
year2019 = home['year']==2019
month2 = home['month']==2
month3 = home['month']==3
month4 = home['month']==4

year2019_ = out['year']==2019
month2_ = out['month']==2
month3_ = out['month']==3
month4_ = out['month']==4

home_2019 = home[year2019&(month2|month3|month4)]
out_2019 = out[year2019_&(month2_|month3_|month4_)]
```

```
#2020년 234월 데이터 뽑아내기
year2020 = home['year']==2020
month2 = home['month']==2
month3 = home['month']==3
month4 = home['month']==4

year2020_ = out['year']==2020
month2_ = out['month']==2
month3_ = out['month']==3
month4_ = out['month']==4

home_2020 = home[year2020&(month2|month3|month4)]
out_2020 = out[year2020_&(month2_|month3_|month4_)]
```

모델 학습 2019년 2,3,4월과 2020년 2,3,4월 차이

```
#기존 데이터의 피처가 같은 것끼리 매칭시키기 위해 merge
home_merge = pd.merge(home_2019,home_2020,on=['CARD_SIDO_NM','STD_CLSS_NM','HOM_SIDO_NM','AGE','FLC','month'])
out_merge = pd.merge(out_2019,out_2020,on=['CARD_SIDO_NM','STD_CLSS_NM','HOM_SIDO_NM','AGE','FLC','month'])

#불필요하게 추가된 열 삭제
home_merge = home_merge.drop(['year_x','year_y','out_home_x','diff_x','검색량(=100)_x','confirmed_x','소비자심리지수_x'],axis=1)
out_merge = out_merge.drop(['year_x','year_y','out_home_x','diff_x','검색량(=100)_x','confirmed_x','소비자심리지수_x'],axis=1)

#2019년 2020년 AMT 차이값 계산
home_merge['AMT_DIFF'] = (home_merge['AMT_x']-home_merge['AMT_y'])
out_merge['AMT_DIFF'] = (out_merge['AMT_x']-out_merge['AMT_y'])

home_merge = home_merge.drop(['AMT_x','AMT_y'],axis=1)
out_merge = out_merge.drop(['AMT_x','AMT_y'],axis=1)
```

```
home_merge
```

	CARD_SIDO_NM	STD_CLSS_NM	HOM_SIDO_NM	AGE	FLC	month	out_home_y	diff_y	검색량(=100)_y	confirmed_y	소비자심리지수_y	AMT_DIFF
0	강원	건강보조식품 소매업	강원	1020s	family	2	1	-0.106782	84.25142	7.0	97.2	-143800
1	강원	건강보조식품 소매업	강원	1020s	single	2	1	-0.106782	84.25142	7.0	97.2	1075900
2	강원	건강보조식품 소매업	강원	3060s	family	2	1	-0.106782	84.25142	7.0	97.2	68946910
3	강원	골프장 운영업	강원	1020s	family	2	1	0.034461	84.25142	7.0	97.2	1267120
4	강원	골프장 운영업	강원	1020s	single	2	1	0.034461	84.25142	7.0	97.2	5332615
...	...	...	...	...	...	...	...	...	...	...	...	...

모델 학습 2019년 2,3,4월과 2020년 2,3,4월 차이

```
#기존 데이터의 피처가 같은 것끼리 매칭시키기 위해 merge
home_merge = pd.merge(home_2019,home_2020,on=['CARD_SIDO_NM','STD_CLSS_NM','HOM_SIDO_NM','AGE','FLC','month'])
out_merge = pd.merge(out_2019,out_2020,on=['CARD_SIDO_NM','STD_CLSS_NM','HOM_SIDO_NM','AGE','FLC','month'])

#불필요하게 추가된 열 삭제
home_merge = home_merge.drop(['year_x','year_y','out_home_x','diff_x','검색량(=100)_x','confirmed_x','소비자심리지수_x'],axis=1)
out_merge = out_merge.drop(['year_x','year_y','out_home_x','diff_x','검색량(=100)_x','confirmed_x','소비자심리지수_x'],axis=1)

#2019년 2020년 AMT 차이값 계산
home_merge['AMT_DIFF'] = (home_merge['AMT_x']-home_merge['AMT_y'])
out_merge['AMT_DIFF'] = (out_merge['AMT_x']-out_merge['AMT_y'])

home_merge = home_merge.drop(['AMT_x','AMT_y'],axis=1)
out_merge = out_merge.drop(['AMT_x','AMT_y'],axis=1)
```

home_merge

	CARD_SIDO_NM	STD_CLSS_NM	HOM_SIDO_NM	AGE	FLC	month	out_home_y	diff_y	검색량(=100)_y	confirmed_y	소비자심리지수_y	AMT_DIFF
0	강원	건강보조식품 소매업	강원	1020s	family	2	1	-0.106782	84.25142	7.0	97.2	-143800
1	강원	건강보조식품 소매업	강원	1020s	single	2	1	-0.106782	84.25142	7.0	97.2	1075900
2	강원	건강보조식품 소매업	강원	3060s	family	2	1	-0.106782	84.25142	7.0	97.2	68946910
3	강원	골프장 운영업	강원	1020s	family	2	1	0.034461	84.25142	7.0	97.2	1267120
4	강원	골프장 운영업	강원	1020s	single	2	1	0.034461	84.25142	7.0	97.2	5332615
...	...	...	...	...	...	...	...	...	...	...	...	...

지역별, 업종별, 연령대별, 가구 생애 주기 별

코로나에 따른 카드사용금액 감소값을 나타냄.

모델 학습 2019년 2,3,4월과 2020년 2,3,4월 차이

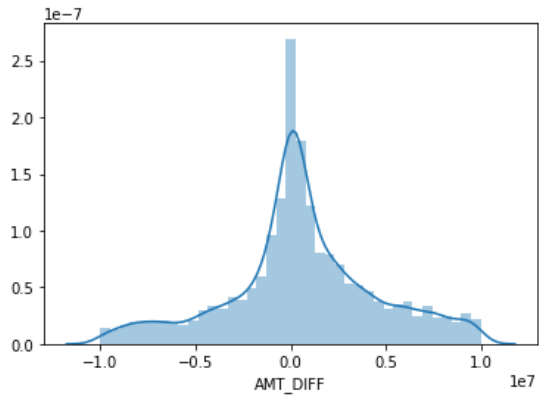
Target이 된 AMT_DIFF

- 분포 조정을 위해 log를 씌우려 했으나 음수 값에 대한 처리 쉽지 않았음
- 이상치 범위 제거한 AMT_DIFF의 분포가 좋았음

```
#이상치 범위 조정
home_merge = home_merge[home_merge['AMT_DIFF']<10000000]
home_merge = home_merge[-10000000<home_merge['AMT_DIFF']]
out_merge = out_merge[out_merge['AMT_DIFF']<1000000]
out_merge = out_merge[-1000000<out_merge['AMT_DIFF']]
```

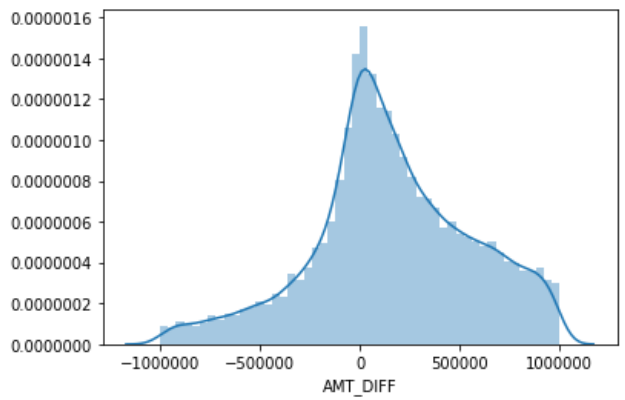
```
#타겟값 형태파악
sns.distplot(home_merge['AMT_DIFF'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x191042b6608>
```




```
#타겟값 형태파악
sns.distplot(out_merge['AMT_DIFF'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x191000359c8>
```



모델 학습

명목형 변수 원핫인코딩

```
#명목형 변수 원핫인코딩
home_dummy = pd.get_dummies(home_merge,columns=['CARD_SIDO_NM','STD_CLSS_NM','HOM_SIDO_NM','AGE','FLC','month'])
out_dummy = pd.get_dummies(out_merge,columns=['CARD_SIDO_NM','STD_CLSS_NM','HOM_SIDO_NM','AGE','FLC','month'])
```

평가 기준인 RMSLE 함수 선언

```
def rmsle(estimator, X, y0):
    pred = estimator.predict(X)
    squared_error = (np.log((y0+1)/(pred+1)))** 2
    rmsle = np.sqrt(np.mean(squared_error))
    if math.isnan(rmsle):
        print("this is a nan")

    return -rmsle
```

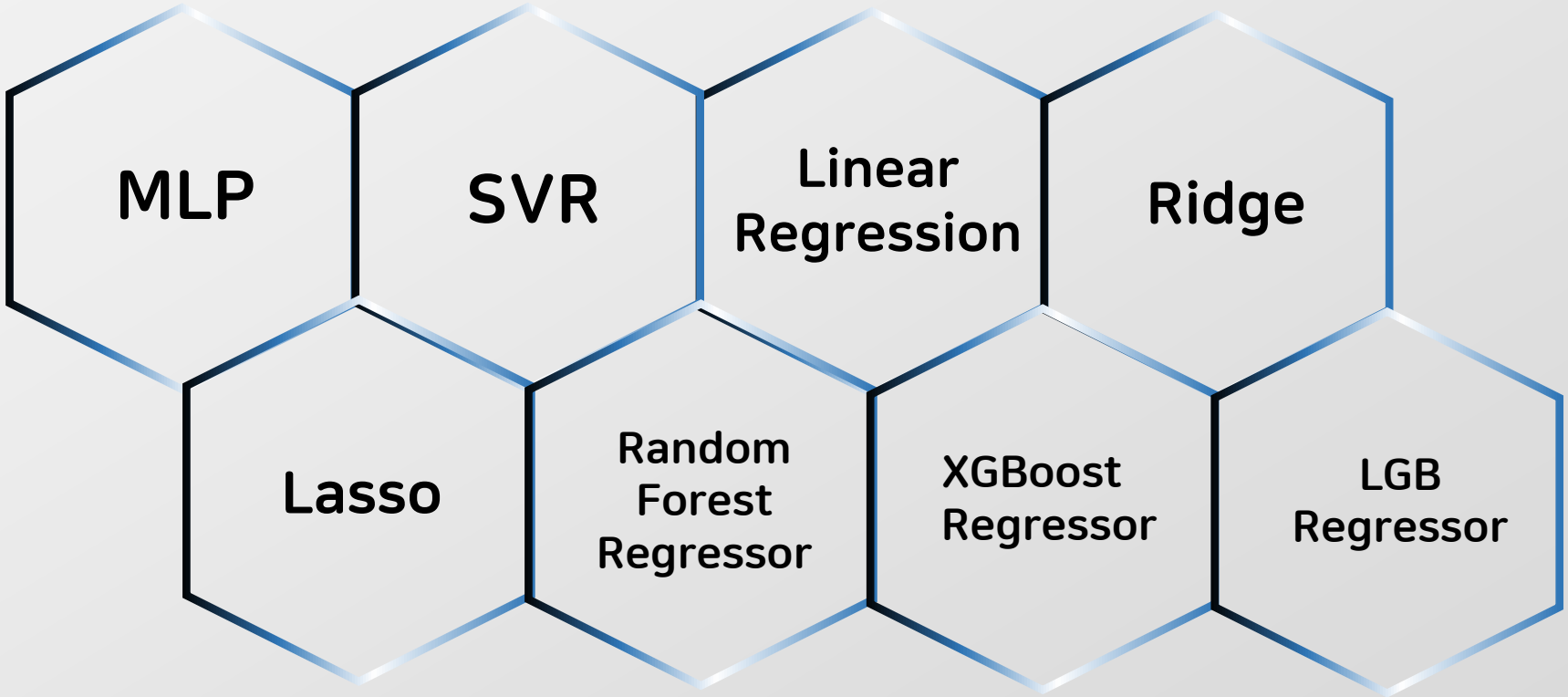
$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1)-\log(a_i+1))^2}$$

모델 학습

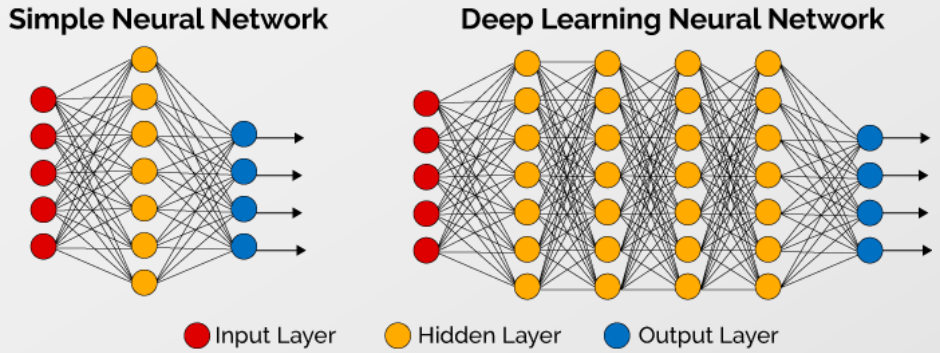
피처, 타겟 선언 & 학습, 테스트 데이터 분리

```
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

#인풋데이터(피처) 생성
home_features = home_dummy.drop('AMT_DIFF',axis=1, inplace=False)
out_features = out_dummy.drop('AMT_DIFF',axis=1, inplace=False)

#타겟값 생성
home_target = home_dummy['AMT_DIFF']
out_target = out_dummy['AMT_DIFF']

#학습,테스트 데이터 분리
Xhome_train, Xhome_test, yhome_train, yhome_test = train_test_split(home_features, home_target, test_size=0.2, random_state=0)
Xout_train, Xout_test, yout_train, yout_test = train_test_split(out_features, out_target, test_size=0.2, random_state=0)
```

모델 학습- 후보 8개


MLP
SVR
Linear Regression
Ridge
Lasso
Random Forest Regressor
XGBoost Regressor
LGB Regressor

MLP (Multilayer Perceptron)



sklearn.neural_network.MLPRegressor 함수로 구현 가능

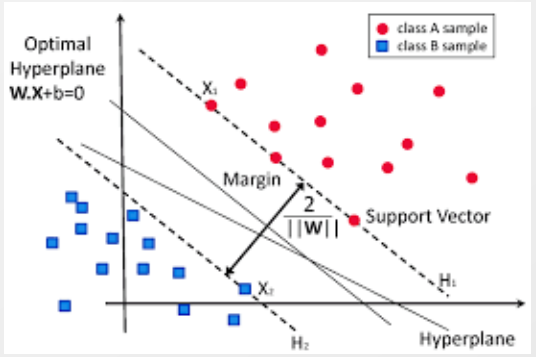
class `sklearn.neural_network.` **`MLPRegressor`**(*hidden_layer_sizes=(100,), activation='relu', *, solver='adam', alpha=0.0001, batch_size='auto', learning_rate='constant', learning_rate_init=0.001, power_t=0.5, max_iter=200, shuffle=True, random_state=None, tol=0.0001, verbose=False, warm_start=False, momentum=0.9, nesterovs_momentum=True, early_stopping=False, validation_fraction=0.1, beta_1=0.9, beta_2=0.999, epsilon=1e-08, n_iter_no_change=10, max_fun=15000*) [source]

Pipeline,
GridSearchCV

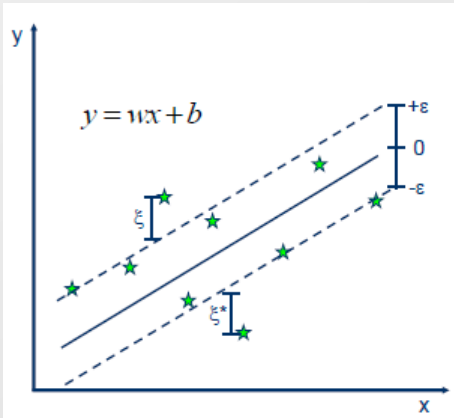
```
pipe_mlp = Pipeline([('scaler', StandardScaler()), ('reg', MLPRegressor(random_state=0))])

param_range = [(100,100),(100,50),(100,50,10),(100, 50, 10, 10, 10),(10, 10, 10, 10, 10)]
param_grid = [{'scaler': [StandardScaler()],'reg__hidden_layer_sizes': param_range}]

mlp_home = GridSearchCV(estimator=pipe_mlp, param_grid=param_grid,
                        scoring=rmsle,cv=5, n_jobs=-1)
mlp_out = GridSearchCV(estimator=pipe_mlp, param_grid=param_grid,
                        scoring=rmsle,cv=5, n_jobs=-1)
```

SVR (Support Vector Regression)

SVM(분류)



SVR(회귀)



- Minimize:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$$

- Constraints:

$$y_i - wx_i - b \le \varepsilon + \xi_i$$
$$wx_i + b - y_i \le \varepsilon + \xi_i^*$$
$$\xi_i, \xi_i^* \ge 0$$

Sklearn.svm.SVR 함수로 구현 가능

class `sklearn.svm.` **SVR**(*, *kernel='rbf', degree=3, gamma='scale', coef0=0.0, tol=0.001, C=1.0, epsilon=0.1, shrinking=True, cache_size=200, verbose=False, max_iter=-1*) [source]

Pipeline,
GridSearchCV

```
pipe_svr = Pipeline([('scaler', StandardScaler()), ('reg', SVR())])

param_range = [0.01, 0.1, 1.0, 10.0]
param_grid = [{'scaler': [StandardScaler()],'reg__C': param_range,
               'reg__kernel': ['linear', 'poly', 'rbf', 'sigmoid'],
               'reg__gamma':['auto','scale']},
              {'scaler': [MinMaxScaler()],'reg__C': param_range,
               'reg__kernel': ['linear', 'poly', 'rbf', 'sigmoid'],
               'reg__gamma':['auto','scale']}]

gs_home = GridSearchCV(estimator=pipe_svr, param_grid=param_grid,
                       scoring=rmsle,cv=5, n_jobs=-1)
gs_out = GridSearchCV(estimator=pipe_svr, param_grid=param_grid,
                      scoring=rmsle,cv=5, n_jobs=-1)
```

최종 결과

	HOME	OUT
MLP	1.33708	1.39085
SVR	1.80264	1.37692
Linear	1.47189	1.56871
Ridge	1.46804	1.30487
Lasso	1.76047	1.31218
Random Forest	1.73928	1.48293
XGBoost	1.78475	1.53596
LGB	2.43378	2.10237

알파에 대한 불확신으로 탈락

HOME은 MLP로 OUT은 SVR로 예측하고 합치기

7월 예측

코로나 확산에 따른 업종별 카드 사용 금액 감소율 학습-> 7월 예측

- 모델을 학습하고 예측하기 위해서 2020년 7월의 코로나 확산세에 대한 데이터 필요
  (대회 규정상, 4월까지의 데이터만 활용해야 했음)

- 코로나 확산세를 나타내는 지표
  - 코로나 검색량
  - 코로나 확진자 수
  - 소비자 심리지수

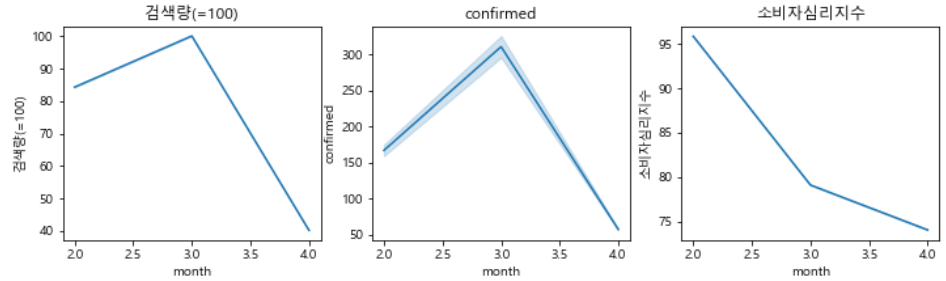
- 7월에 대한 위 3가지의 지표 값 필요.

모델 구축-코로나 확산 수준

2,3,4월 지표 추세

```
# 2020 2월, 3월, 4월 데이터 추출
sub_data = data.loc[((data["month"]==2)&(data["year"]==2020)) | ((data["month"]==3)&(data["year"]==2020))|((data["month"]==4)&(data["year"]==2020)),:]
```

```
# 각 경제지표 추세 확인
plt.figure(figsize = (12,3))
for idx, y in enumerate(sub_data.columns[-3:]):
    plt.subplot(1,3,idx+1)
    sns.lineplot(x="month", y = y,data =sub_data)
    plt.title(y)
```



2,3,4월 지표 값

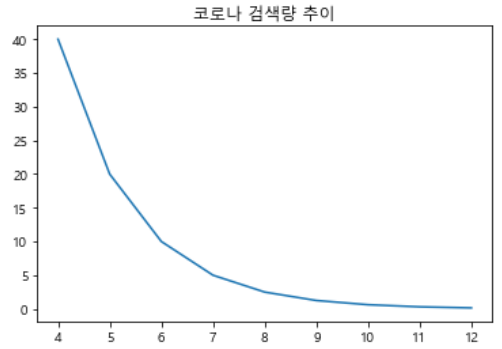
month	검색량(=100)	confirmed	소비자심리지수
2	84.25142	166.912526	95.849880
3	100.00000	311.182177	79.091874
4	40.21097	56.966629	74.031100

반박의 여지가 있지만, 대회 규정상 직관에 의한 선택
-코로나의 확산 수준이 완화된다고 생각
-12월 종식으로 두고 감소해가는 것으로 7월 지표 만들기

모델 구축-코로나 확산 수준

코로나 검색량

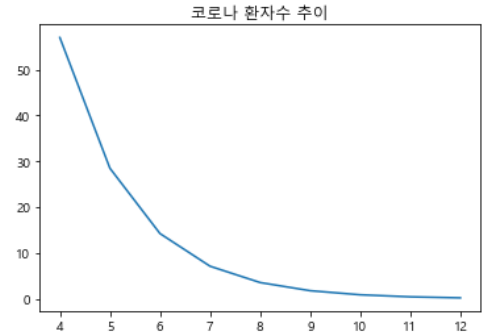
```
# 코로나 검색량 추이 생성
x = np.arange(4,13,1)
y = []
r = 0.5 # 50% 씩 감소할 것으로 예측
for i in range(0,9):
    num = 40*(1-r)**i
    y.append(num)

sns.lineplot(x=x,y=y)
plt.title("코로나 검색량 추이")
plt.show()

# 4~12월 검색량 예측치
corona_search = y.copy()
```




```
corona_search
[40.0, 20.0, 10.0, 5.0, 2.5, 1.25, 0.625, 0.3125, 0.15625]
```

2,3,4월 지표 값

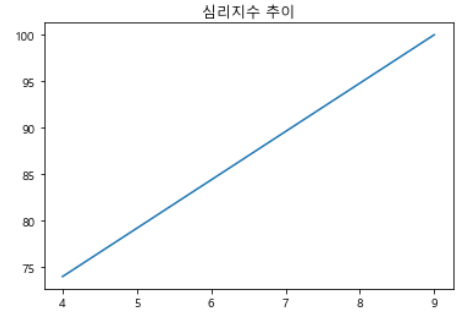
```
# 코로나 환자수 추이 생성
x = np.arange(4,13,1)
y = []
r = 0.5

for i in range(0,9):
    num = 57*(1-r)**i
    y.append(num)

sns.lineplot(x=x,y=y)
plt.title("코로나 환자수 추이")
plt.show()

# 4~12월 검색량 예측치
corona_patient = y.copy()
```




```
corona_patient
[57.0, 28.5, 14.25, 7.125, 3.5625, 1.78125, 0.890625, 0.4453125, 0.22265625]
```

2,3,4월 지표 추세

```
# 소비자 심리지수 추이 생성
# 코로나가 장기화되면서 지친 소비자들 생김-> 9월쯤 소비자 심리지수가 회복될 것
x = np.arange(4,10,1)
y = np.linspace(74,100,6)
r = 0.5 # 증가율!

sns.lineplot(x=x,y=y)
plt.title("심리지수 추이")
plt.show()

# 4~12월 검색량 예측치
consumer_mind = y.copy()
```



모델 구축-코로나 확산 수준

코로나 검색량

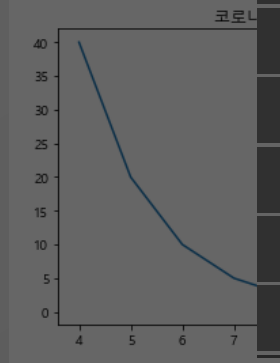
2,3,4월 지표 값

2,3,4월 지표 추세

```
# 코로나 검색량 추이 생성
x = np.arange(4,13,1)
y = []
r = 0.5 # 50% 씩 감소할 것으로 예측
for i in range(0,9):
    num = 40*(1-r)**i
    y.append(num)

sns.lineplot(x=x,y=y)
plt.title("코로나 검색량
plt.show()

# 4~12월 검색량 예측치
corona_search = y.copy()
```

```
# 코로나 환자수 추이 생성
x = np.arange(4,13,1)
y = []
r = 0.5

for i in range(0,9):
```

```
# 소비자 심리지수 추이 생성
# 코로나가 장기화되면서 지친 소비자들 생각-> 9월쯤 소비자 심리지수가 회복될 것
x = np.arange(4,10,1)
y = np.linspace(74,100,6)
r = 0.5 # 증가율!
```



	A	B	C	D
1		search	confirmed	consumer_mind
2	2	84.25142	166.9125259	95.84987962
3	3	100	311.1821773	79.09187396
4	4	40.21097	56.96662864	74.03110026
5	5	20	28.5	79.2
6	6	10	14.25	84.4
7	7	5	7.125	89.6

```
corona_search
[40.0, 20.0, 10.0, 5.0, 2.5, 1.25, 0.625, 0.3125, 0.15625]
```

```
corona_patient
[57.0, 28.5, 14.25, 7.125, 3.5625, 1.78125, 0.890625, 0.4453125, 0.22265625]
```

최종 예측

앞서 예측한 7월 코로나 확산을 나타내는 지표를 추가한 후,
선택한 모델로 7월 최종 예측

```
#학습할 때 사용한 정규화 객체로 테스트 데이터도 fitting
test_home_scaled = scaler1.transform(test_home)
test_out_scaled = scaler2.transform(test_out)

#home은 mlp, out은 svr로 예측
home_pred = mlp_home.predict(test_home_scaled)
out_pred = svr_out.predict(test_out_scaled)
```

AMT_DIFF 값 예측

```
#19년 7월 데이터에서 예측한 값 빼주기(20년 7월 데이터값 예측)
home_77['AMT'] = home_77['AMT']-home_pred
out_77['AMT'] = out_77['AMT']-out_pred
```

2020년 7월 계산

2019년 7월 — home, out_pred = 2020년 7월

최종 결론

한계점

- 대회 규정상, 4월 이전의 데이터만 사용할 수 있었기에 많은 어려움을 겪음
- 코로나와 같은 감염병에 대한 경제 회복 수준에 대한 사전 데이터 존재 X
- 7월 코로나를 순전히 감으로 예측해야 했음
- 주변 지인들이 제주도 여행을 많이 가서 조금 더 긍정적으로 평가했던 직관의 문제

늦참

기본 IDEA

모델 학습

그리드 서치

예측

최종 결과

의의

- 실제 카드 데이터를 기반으로 처음부터 끝까지의 과정을 실행할 수 있었음
- 준비기간의 약 50% 이상을 차지한 EDA과정의 중요성과 그 힘을 깨달음
  - 변수 선택
  - 변수 파생(관광지 구분)
  - 모델 구현 방식(HOME+OUT)

  처음부터 끝까지 큰 틀을 잡는 기준!!!
- 주어진 데이터 외에 코로나가 일상에 미치는 수준을 나타내는 변수를 찾아보고 유의미하게 사용