모두의 주차장 분석 및 예측 모델 프로젝트

TEAM2
유현준(팀장), 김기성, 김한빈, 맹광국, 박선하,
정상현, 정서현, 최창효, 최현숙

목차

모두의 주차장 소개

모두의 주차장 소개

모두가 사용하는
쉽고 편리한 주차 플랫폼

모두의주차장

주차장 정보 안내

- 서울 및 광역시 내 공영/민영/부설주차장 정보 제공
- 유료 주차장의 무료개방 시간을 실시간으로 반영해 안내
- 네비게이션 앱 연동 (Tmap, 네이버 지도, 카카오네비, 원내비, 맵피, 아틀란 3D 지원)
- 무료주차공간 정보 제공

유휴 주차공간 공유

- 거주자 우선 주차, 사무건물, 빌라, 교회 등 주차면의 쓰지 않는 시간을 다른 운전자들에게 공유
- 공유할 시간(요일 및 시간)을 자유롭게 설정 가능
- '21년 2월 기준' 서울시 내 23개, 부산시 8개 자치구, 부천시와 주차공유사업 업무협약 체결

1. 데이터 살펴보기

1. 데이터 살펴보기

실전 DB 데이터 - 주 데이터

	USER_ID	JOIN_DATE	D_TYPE	STORE_ID	GOODS_TYPE	DATE	COUNT	AD1
0	2858	2014-01-07	AA	1892	A	2020-01-01	1	GN
1	5647	2014-02-14	BB	182009	A	2020-01-01	1	J
2	33314	2014-11-20	BB	82431	A	2020-01-01	1	SC
3	37001	2014-12-04	BB	725	C	2020-01-01	1	MP
4	37819	2014-12-07	AA	220691	C	2020-01-01	1	JRR

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 879271 entries, 0 to 879270
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   USER_ID      879271 non-null  int64
 1   JOIN_DATE    879271 non-null  object
 2   D_TYPE       879271 non-null  object
 3   STORE_ID     879271 non-null  int64
 4   GOODS_TYPE   879271 non-null  object
 5   DATE         879271 non-null  object
 6   COUNT        879271 non-null  int64
 7   AD1          879271 non-null  object
dtypes: int64(3), object(5)
memory usage: 53.7+ MB
```

1. 데이터 살펴보기

외부 데이터 - 2020 서울 교통량

	DATE	지점명	지점번호	방향	구분	0시	1시	2시	3시	4시	...	14시	15시	16시	17시	18시	19시	20시	21시	22시	23시
0	2020-01-01	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	712.0	645.0	437.0	309.0	290.0	...	1472.0	1416.0	1483.0	1329.0	1157.0	1014.0	954.0	849.0	780.0	480.0
1	2020-01-02	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	315.0	222.0	186.0	165.0	266.0	...	1792.0	1897.0	1842.0	2061.0	1994.0	1443.0	1233.0	1165.0	1094.0	852.0
2	2020-01-03	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	632.0	457.0	295.0	236.0	279.0	...	2004.0	1929.0	2049.0	2140.0	2178.0	1654.0	1356.0	1260.0	1253.0	941.0
3	2020-01-04	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	740.0	518.0	388.0	331.0	330.0	...	1837.0	1788.0	1588.0	1669.0	1530.0	1222.0	1143.0	1089.0	1039.0	791.0
4	2020-01-05	성산로(금화터널)	A-01	유입	봉원고가차도->독립문역	533.0	424.0	297.0	230.0	209.0	...	1634.0	1673.0	1494.0	1429.0	1288.0	1035.0	987.0	884.0	803.0	564.0

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 98820 entries, 0 to 98819
Data columns (total 29 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   DATE    98820 non-null  datetime64[ns]
 1   지점명     98820 non-null  object
 2   지점번호    98820 non-null  object
 3   방향      98820 non-null  object
 4   구분      98820 non-null  object
 5   0시      91985 non-null  float64
 6   1시      91962 non-null  float64
 7   2시      91960 non-null  float64
 8   3시      91979 non-null  float64
 9   4시      91999 non-null  float64
 10  5시      92036 non-null  float64
 11  6시      92079 non-null  float64
 12  7시      92095 non-null  float64
 13  8시      92081 non-null  float64
 14  9시      92110 non-null  float64
 15  10시     92144 non-null  float64
 16  11시     92100 non-null  float64
 17  12시     92067 non-null  float64
 18  13시     92032 non-null  float64
 19  14시     92053 non-null  float64
 20  15시     92066 non-null  float64
 21  16시     92093 non-null  float64
 22  17시     92067 non-null  float64
 23  18시     92079 non-null  float64
 24  19시     92034 non-null  float64
 25  20시     92040 non-null  float64
 26  21시     92023 non-null  float64
 27  22시     91824 non-null  float64
 28  23시     91698 non-null  float64
dtypes: datetime64[ns](1), float64(24), object(4)
memory usage: 21.9+ MB
```

1. 데이터 살펴보기

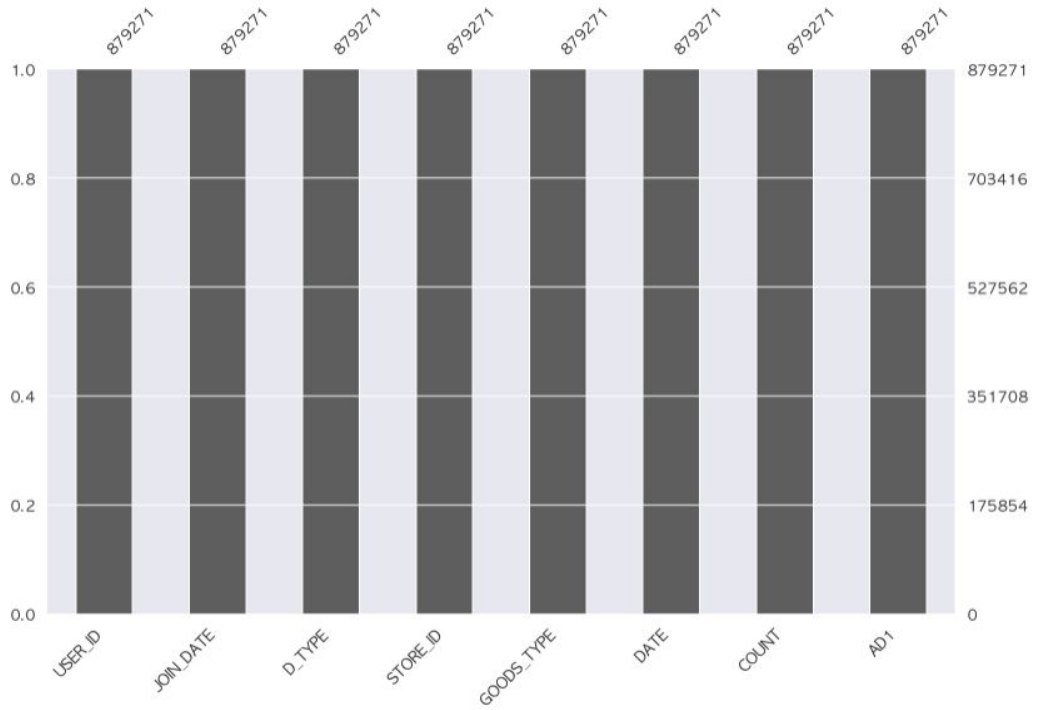
외부 데이터 - 서울시 기상데이터

	지점	일시	기온	강수량	풍속	습도	일조시간	적설량	지면온도	지중온도
0	108	2020-01-01 1:00	-5.9	NaN	1.7	40	NaN	NaN	-2.4	3.2
1	108	2020-01-01 2:00	-5.7	NaN	0.1	42	NaN	NaN	-2.4	3.1
2	108	2020-01-01 3:00	-5.6	0.0	0.0	46	NaN	NaN	-2.7	3.1
3	108	2020-01-01 4:00	-5.4	NaN	0.0	50	NaN	NaN	-2.5	3.0
4	108	2020-01-01 5:00	-5.2	NaN	0.0	55	NaN	NaN	-2.2	3.0

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8760 entries, 0 to 8759
Data columns (total 10 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   지점      8760 non-null   int64
 1   일시      8760 non-null   object
 2   기온      8759 non-null   float64
 3   강수량     1059 non-null   float64
 4   풍속      8760 non-null   float64
 5   습도      8760 non-null   int64
 6   일조시간    4791 non-null   float64
 7   적설량     208 non-null    float64
 8   지면온도    8752 non-null   float64
 9   지중온도    8748 non-null   float64
dtypes: float64(7), int64(2), object(1)
memory usage: 684.5+ KB
```

1. 데이터 살펴보기 - 결측치 확인

결측치 확인 - 결측치 없음




```
USER_IDhas 0 null values.

JOIN_DATEhas 0 null values.

D_TYPEhas 0 null values.

STORE_IDhas 0 null values.

GOODS_TYPEhas 0 null values.

DATEhas 0 null values.

COUNThas 0 null values.

AD1has 0 null values.
```

2. 데이터 탐색 - EDA

2. 데이터 탐색 - 1) 각 컬럼 의미 탐색

JOIN DATE - 가입일

1970년 ~ 2020년 데이터

그래프가 우상향함

가입자가 꾸준했으며, 가입자가 증가하고 있음을 알 수 있음.

2. 데이터 탐색 - 1) 각 컬럼 의미 탐색

JOIN DATE - 가입일

	USER_ID	JOIN_DATE	JOIN_YEAR	JOIN_MONTH	JOIN_DAY	D_TYPE	STORE_ID	GOODS_TYPE	DATE	PAY_YEAR	PAY_MONTH	PAY_DAY	WEEKDAY	COUNT	AD1
437	1081430	1970-01-01	1970	1	1	CC	90070	A	2020-01-01	2020	1	1	Wed	1	GD
2196	1410151	1970-01-01	1970	1	1	CC	92437	B	2020-01-02	2020	1	2	Thu	1	J
2204	1415023	1970-01-01	1970	1	1	CC	131081	A	2020-01-02	2020	1	2	Thu	1	YO
2434	125582	1970-01-01	1970	1	1	CC	91885	A	2020-01-03	2020	1	3	Fri	1	YD
2645	602188	1970-01-01	1970	1	1	CC	2334	B	2020-01-03	2020	1	3	Fri	1	MP

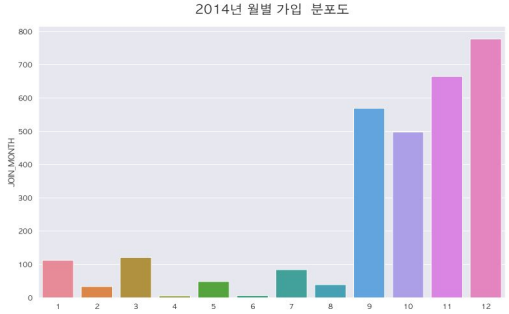
가입년도와 동떨어져 보이는 1970년도가 1093건 존재함

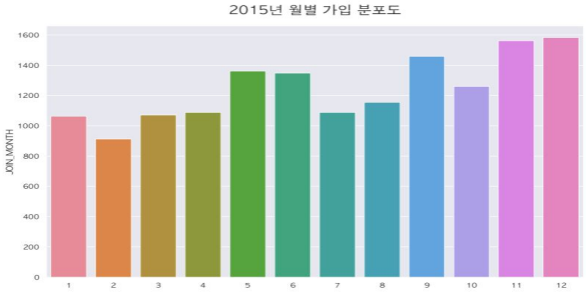
1970년도 모두 가입일이 1970년 1월 1일로 되어 있음

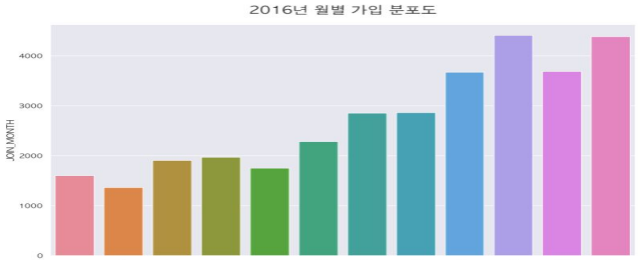
가입일 기준은 2013년 부터 2020년도의 기간이 확인 됨 (런칭일 2013년 12월)

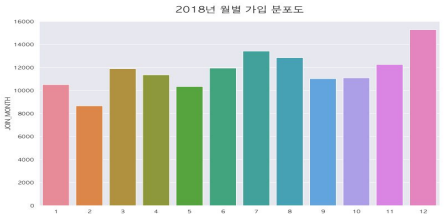
2. 데이터 탐색 - 1) 각 컬럼 의미 탐색

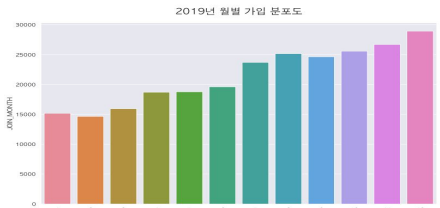
JOIN DATE - 가입일

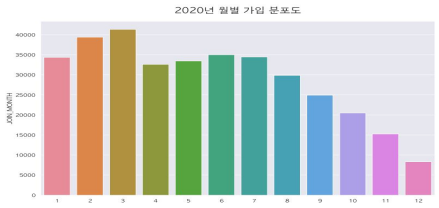






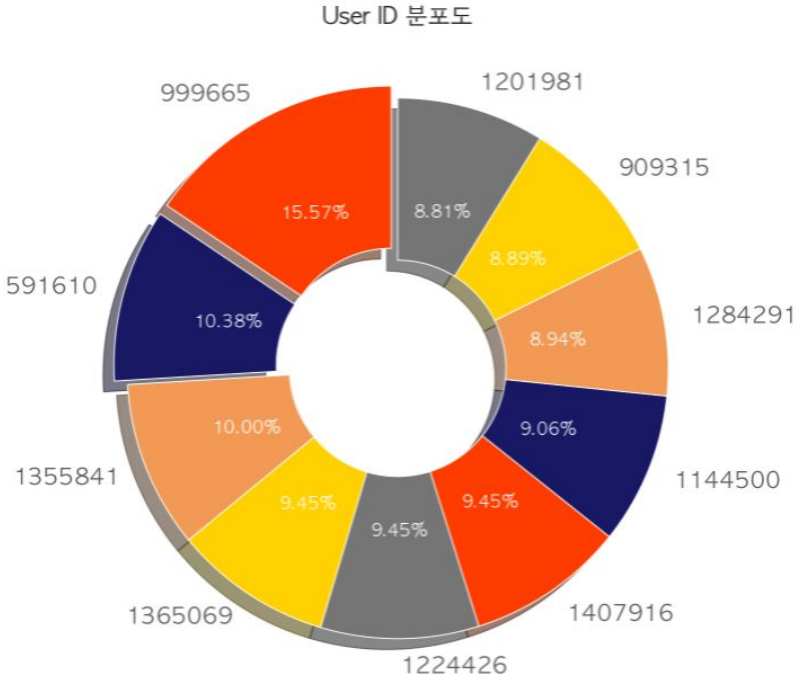






2. 데이터 탐색 - 1) 각 컬럼 의미 탐색

USER_ID - 유저 ID 정보

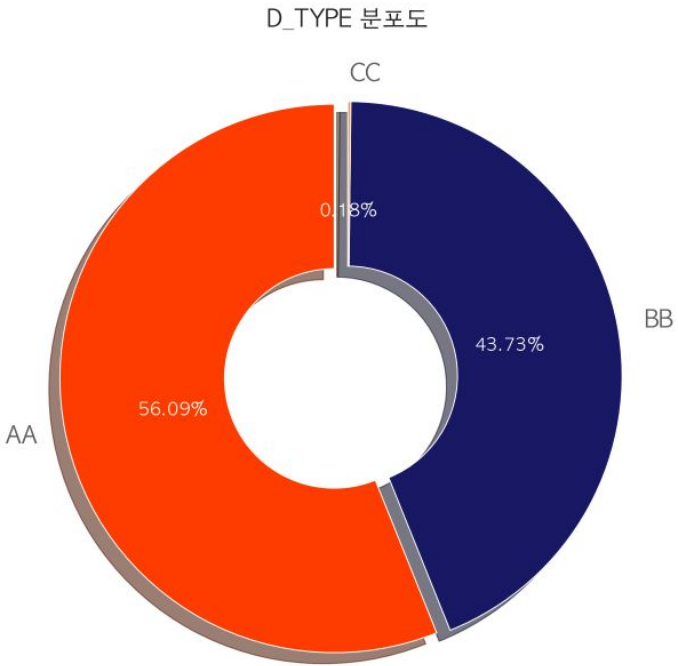


	USER_ID	JOIN_DATE	JOIN_YEAR	JOIN_MONTH	JOIN_DAY	D_TYPE	STORE_ID	GOODS_TYPE	DATE	PAY_YEAR	PAY_MONTH	PAY_DAY	WEEKDAY	COUNT	AD1
372	999665	2018-12-16	2018	12	16	CC	82399	A	2020-01-01	2020	1	1	Wed	6	JRR
1527	999665	2018-12-16	2018	12	16	CC	104988	A	2020-01-02	2020	1	2	Thu	11	GN
3121	999665	2018-12-16	2018	12	16	CC	181832	A	2020-01-03	2020	1	3	Fri	7	SC
5034	999665	2018-12-16	2018	12	16	CC	109223	A	2020-01-04	2020	1	4	Sat	28	MP
6919	999665	2018-12-16	2018	12	16	CC	104916	A	2020-01-05	2020	1	5	Sun	14	GN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
867260	999665	2018-12-16	2018	12	16	CC	2428	A	2020-12-27	2020	12	27	Sun	17	CY
868847	999665	2018-12-16	2018	12	16	CC	109267	A	2020-12-28	2020	12	28	Mon	24	J
871562	999665	2018-12-16	2018	12	16	CC	90193	C	2020-12-29	2020	12	29	Tue	23	GW
874317	999665	2018-12-16	2018	12	16	CC	220797	D	2020-12-30	2020	12	30	Wed	40	GS
877120	999665	2018-12-16	2018	12	16	CC	220893	A	2020-12-31	2020	12	31	Thu	42	J

366 rows × 15 columns

2. 데이터 탐색 - 1) 각 컬럼 의미 탐색

D_TPYE - 정보없음



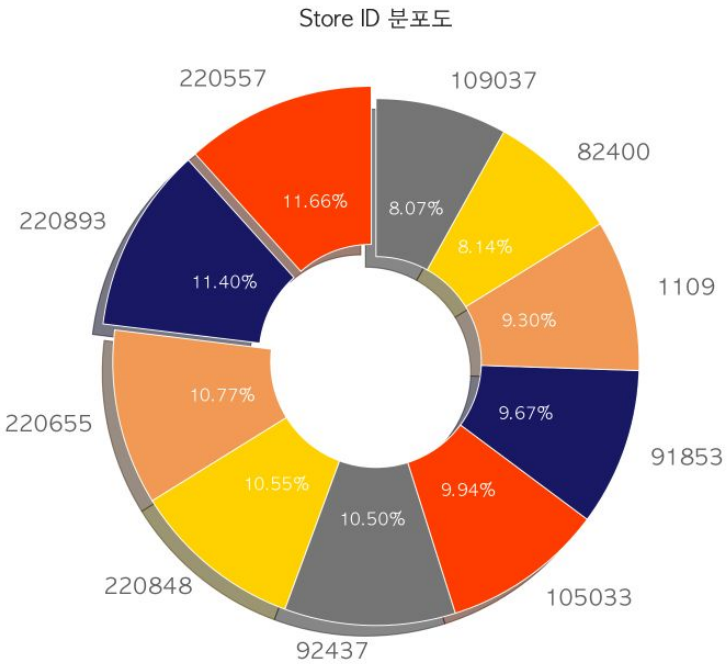
AA: 56.09%(493,166건), BB: 43.73%(384,541건), CC: 0.18%(471건)

USER_ID 별로 D_TYPE의 개수를 채크해보니 두 컬럼간 1대1로 매칭이 됨.

D_TYPE이 회원정보과 관련이 있는 데이터 즉 회원의 등급이지 않을까 추측해봄

2. 데이터 탐색 - 1) 각 컬럼 의미 탐색

STORE ID - 주차장 ID 정보

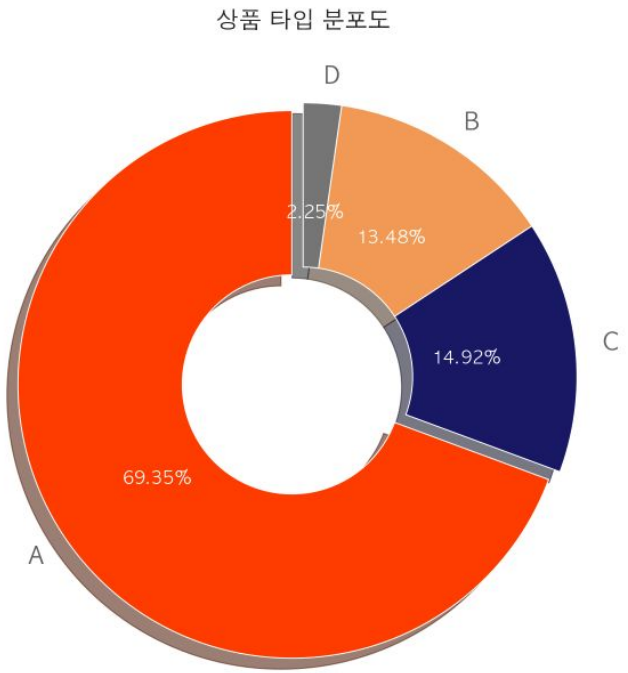


Store ID는 220557과 220893의 ID가 가장 많았다.

해당 데이터에 주소 데이터를 대조하여 살펴보니 각각 용산지역의 주차장, 중구 지역의 주차장임이 나타났다.

2. 데이터 탐색 - 1) 각 컬럼 의미 탐색

GOODS_TYPE - 정보 없음

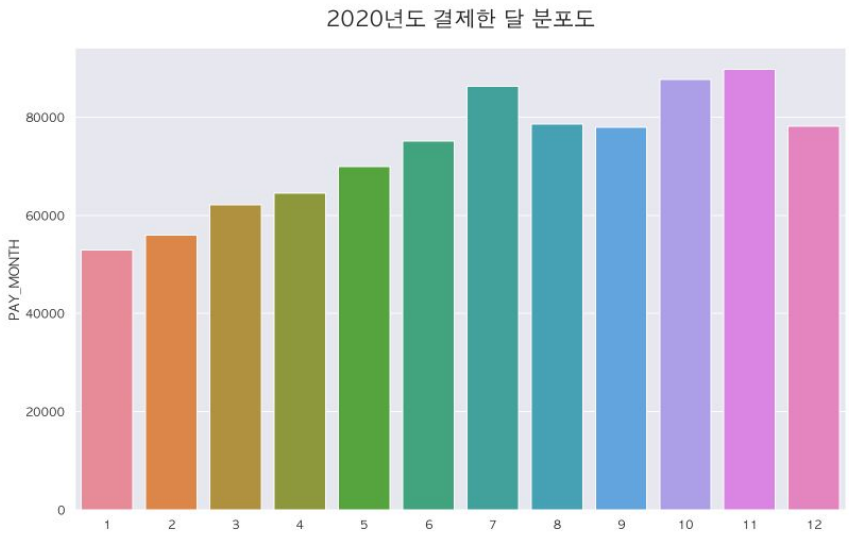


GOOD_TYPE에서 A타입이 69.35%(609,790건), C타입이 14.92%(131,163건), B타입이 13.48%(118,541건), D타입이 2.25%(19,777건) 순이다.

A 타입이 가장 큰 비중을 차지하며, C 타입과 B 타입은 비슷한 수준이고, D 타입은 가장 적은 비중을 차지함을 확인함.

2. 데이터 탐색 - 1) 각 컬럼 의미 탐색

DATE - 결제일



해당 데이터의 결제일을 보면 모두 2020년도 자료만 존재한다.

결제달을 분석해보니 가장 많은 달은 11월과 7월 이다.

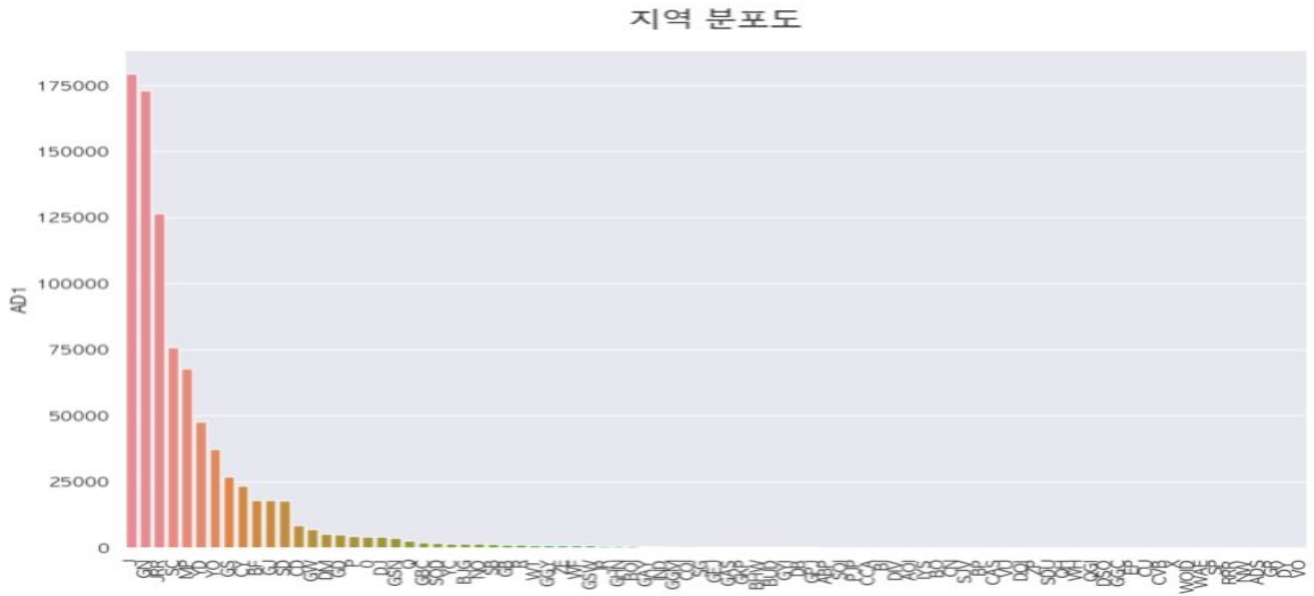
2. 데이터 탐색 - 1) 각 컬럼 의미 탐색

COUNT - 결제건수

결제건수는 1건이 가장 많긴 하지만, 다양한 유형의 결제건수가 함께 존재함을 확인함

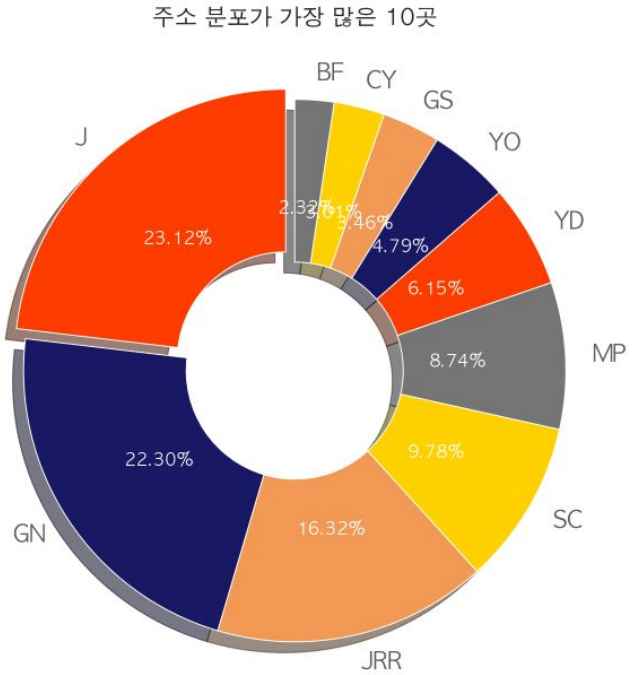
2. 데이터 탐색 - 1) 각 컬럼 의미 탐색

AD1 - 주차장 주소



2. 데이터 탐색 - 1) 각 컬럼 의미 탐색

AD1 - 주차장 주소



2. 데이터 탐색 - 2) 예측 모델을 위한 주요 Feature 선별을 위한 분석

먼저 가입일 1970년과 999665 ID 제거

	USER_ID	JOIN_DATE	JOIN_YEAR	JOIN_MONTH	JOIN_DAY	D_TYPE	STORE_ID	GOODS_TYPE	DATE	PAY_YEAR	PAY_MONTH	PAY_DAY	WEEKDAY	COUNT	AD1
437	1081430	1970-01-01	1970	1	1	CC	90070	A	2020-01-01	2020	1	1	Wed	1	GD
2196	1410151	1970-01-01	1970	1	1	CC	92437	B	2020-01-02	2020	1	2	Thu	1	J
2204	1415023	1970-01-01	1970	1	1	CC	131081	A	2020-01-02	2020	1	2	Thu	1	YO
2434	125582	1970-01-01	1970	1	1	CC	91885	A	2020-01-03	2020	1	3	Fri	1	YD
2645	602188	1970-01-01	1970	1	1	CC	2334	B	2020-01-03	2020	1	3	Fri	1	MP
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
866536	1723400	1970-01-01	1970	1	1	CC	190188	A	2020-12-26	2020	12	26	Sat	1	JR
867913	1723400	1970-01-01	1970	1	1	CC	190188	A	2020-12-27	2020	12	27	Sun	1	JR
868210	1825656	1970-01-01	1970	1	1	CC	219936	A	2020-12-27	2020	12	27	Sun	1	YD
873707	1828010	1970-01-01	1970	1	1	CC	220635	C	2020-12-29	2020	12	29	Tue	1	GN
879171	1829458	1970-01-01	1970	1	1	CC	137452	A	2020-12-31	2020	12	31	Thu	1	CY

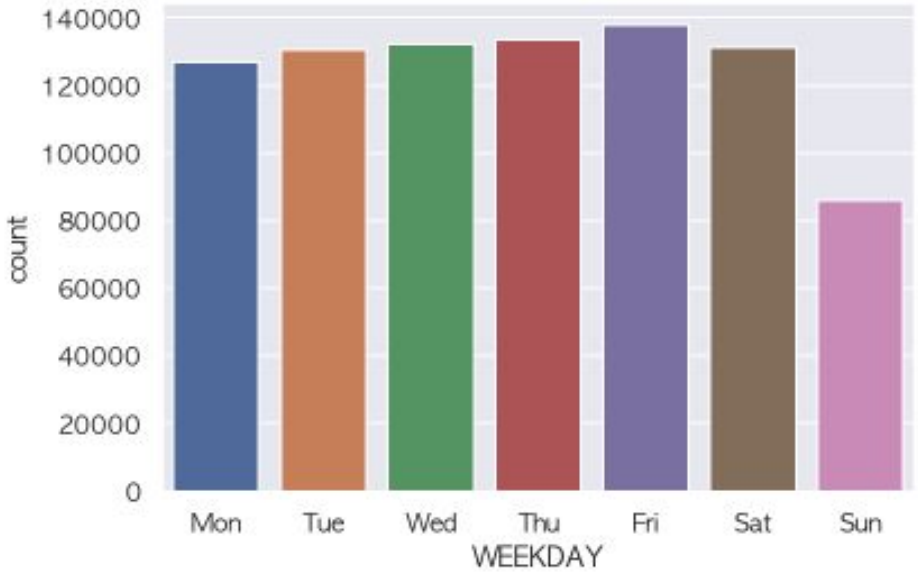
1093 rows × 15 columns

	USER_ID	JOIN_DATE	JOIN_YEAR	JOIN_MONTH	JOIN_DAY	D_TYPE	STORE_ID	GOODS_TYPE	DATE	PAY_YEAR	PAY_MONTH	PAY_DAY	WEEKDAY	COUNT	AD1
372	999665	2018-12-16	2018	12	16	CC	82399	A	2020-01-01	2020	1	1	Wed	6	JRR
1527	999665	2018-12-16	2018	12	16	CC	104988	A	2020-01-02	2020	1	2	Thu	11	GN
3121	999665	2018-12-16	2018	12	16	CC	181832	A	2020-01-03	2020	1	3	Fri	7	SC
5034	999665	2018-12-16	2018	12	16	CC	109223	A	2020-01-04	2020	1	4	Sat	28	MP
6919	999665	2018-12-16	2018	12	16	CC	104916	A	2020-01-05	2020	1	5	Sun	14	GN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
867260	999665	2018-12-16	2018	12	16	CC	2428	A	2020-12-27	2020	12	27	Sun	17	CY
868847	999665	2018-12-16	2018	12	16	CC	109267	A	2020-12-28	2020	12	28	Mon	24	J
871562	999665	2018-12-16	2018	12	16	CC	90193	C	2020-12-29	2020	12	29	Tue	23	GW
874317	999665	2018-12-16	2018	12	16	CC	220797	D	2020-12-30	2020	12	30	Wed	40	GS
877120	999665	2018-12-16	2018	12	16	CC	220893	A	2020-12-31	2020	12	31	Thu	42	J

366 rows × 15 columns

2. 데이터 탐색 - 2) 예측 모델을 위한 주요 Feature 선별을 위한 분석

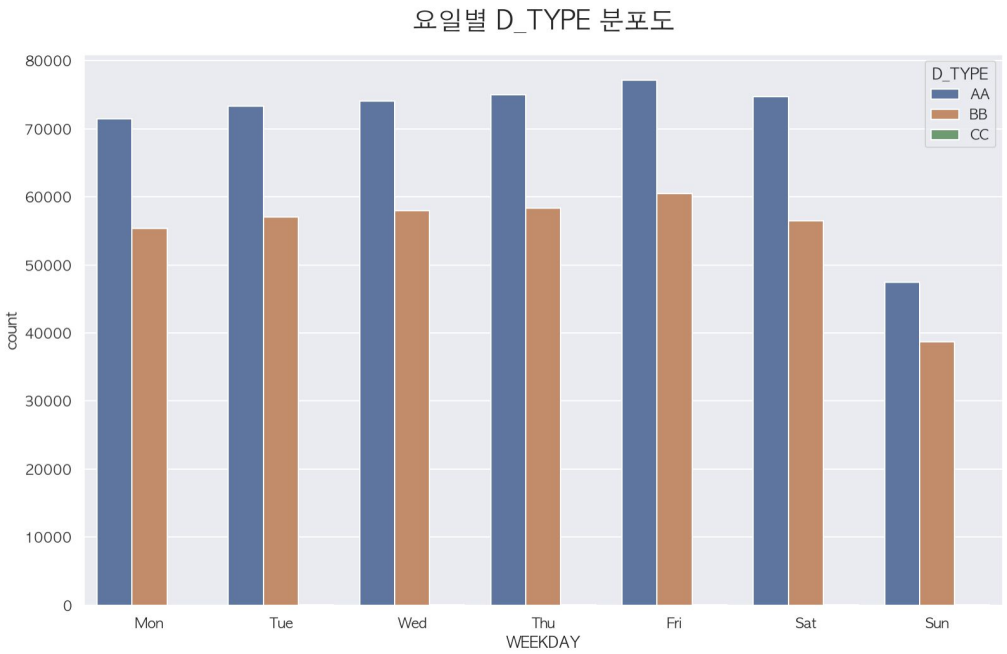
요일 및 날짜에 따른 이용량 변화 분석



2. 데이터 탐색 - 2) 예측 모델을 위한 주요 Feature 선별을 위한 분석

특정 Feature 기준 분류 후, 결제 일자별, 요일별 이용량 추이 분석

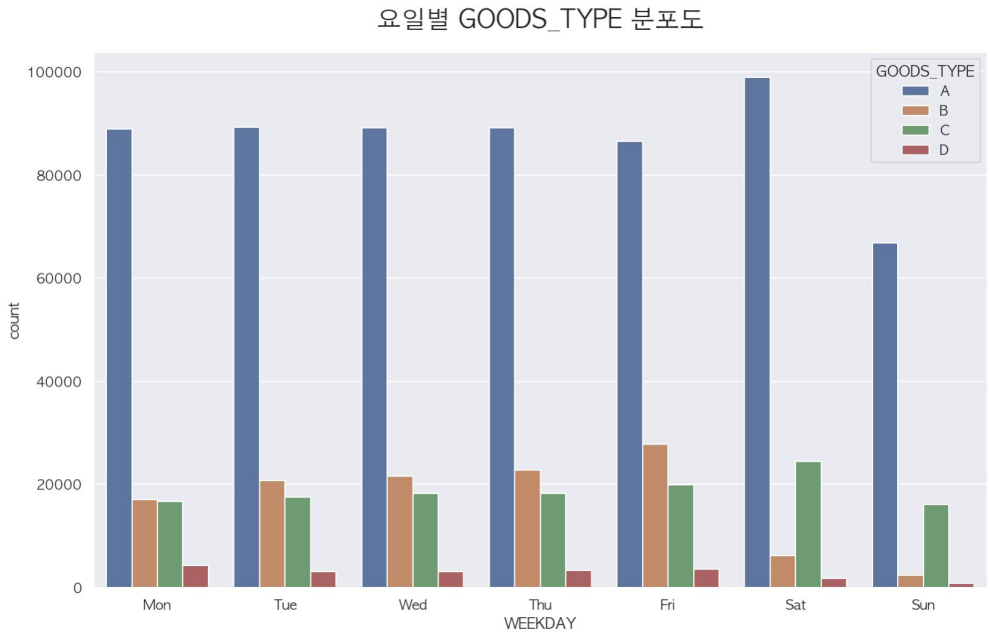
-> D_TYPE 기준 분류



2. 데이터 탐색 - 2) 예측 모델을 위한 주요 Feature 선별을 위한 분석

특정 Feature 기준 분류 후, 결제 일자별, 요일별 이용량 추이 분석

-> GOODS_TYPE기준 분류



2. 데이터 탐색 - 2) 예측 모델을 위한 주요 Feature 선별을 위한 분석

특정 Feature 기준 분류 후, 결제 일자별, 요일별 이용량 추이 분석

-> 이용(결제)월 기준 분류

2. 데이터 탐색 - 2) 예측 모델을 위한 주요 Feature 선별을 위한 분석

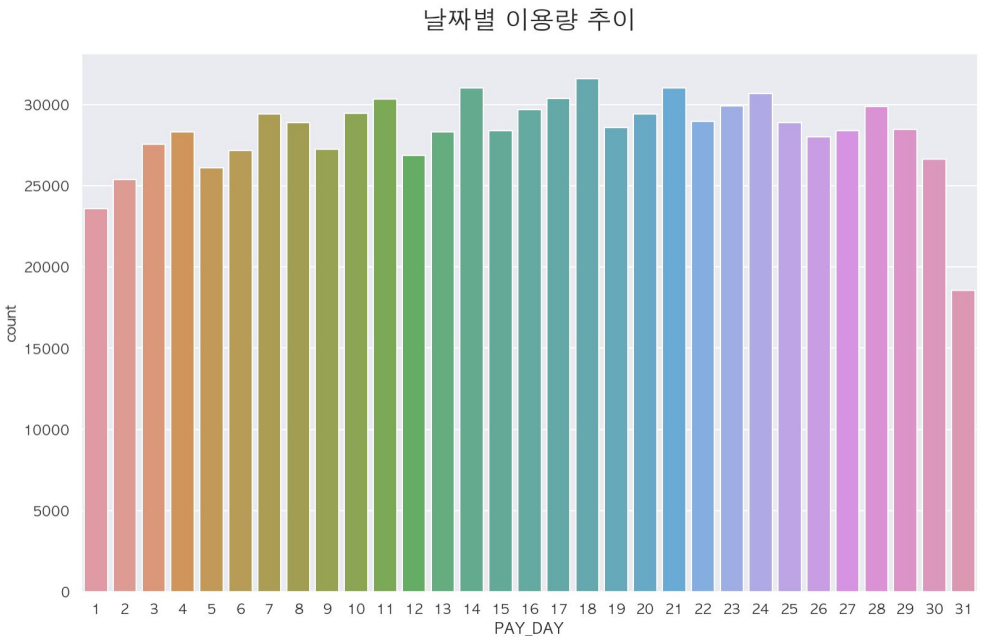
특정 Feature 기준 분류 후, 결제 일자별, 요일별 이용량 추이 분석

-> 월별/GOODS_TYPE 별 복합 적용

2. 데이터 탐색 - 2) 예측 모델을 위한 주요 Feature 선별을 위한 분석

특정 Feature 기준 분류 후, 결제 일자별, 요일별 이용량 추이 분석

-> 날짜별 이용량 추이 분석



2. 데이터 탐색 - 2) 예측 모델을 위한 주요 Feature 선별을 위한 분석

정리

- 요일에 따른 이용량 변화는 물론이고, 월별, 날짜별, "GOODS_TYPE"별로도 이용량이 서로 다른 특성을 보임을 확인되었다.

- 다만, 현재 주어진 예측모델 적용 기준상 Traing 데이터(1-9월 기준)와 Test 데이터(10-12월 기준)의 기준월이 상이하므로, 예측모델 적용 시 "월" Feature는 제외시키기로 함

3. 모델링 구축 및 검증

3. 모델링 구축 및 검증 - 1) 모델링에 필요한 Feature

Feature 별 적용여부 판단

1. JOIN_DATE -> 유저 가입일 : 유저별 1개씩 존재하는 가입일 정보로 USER_ID대비 추가 변별력이 없어보임 - 적용 보류
2. D_TYPE -> 회원 등급(추정): 유저별 1개씩 존재하는 값으로 USER_ID대비 추가로 변별력이 없어보임 - 적용 보류
3. STORE_ID -> 등록 주차장 ID: 유저가 이용한 점포의 고유 id로 변별력 있어보임 - 적용
4. GOODS_TYPE -> 주차권 타입(추정): 각 TYPE별로 이용행태가 다르게 나타나므로 변별력이 있어보임 - 적용
5. DATE -> 날짜별로 이용 행태가 다르게 나타나므로 변별력 있어 보임 - 적용 / 단, 연도 및 월을 제외한 이용(결제)일("DATE_D") 기준으로 적용
6. AD1 -> 점포id별로 1개씩 존재하는 주소값으로 STORE_ID 대비 추가로 변별력이 없어 보임 - 적용 보류
7. 추가적용 Feature: WEEKDAY -> DATE에서 추출한 요일 정보로 요일별 특성 반영이 필요해 보임.

-> "STORE_ID", "GODDS_TYPE", "DATE", "WEEKDAY"를 기준으로 예측 모델 적용

3. 모델링 구축 및 검증 - 2) Label Encoding 후 성능측정

Label Encoding 후 필요한 Feature만 추출

JOIN_DATE	JOIN_YEAR	JOIN_MONTH	JOIN_DAY	D_TYPE	STORE_ID	GOODS_TYPE	DATE	PAY_YEAR	PAY_MONTH	PAY_DAY	WEEKDAY	COUNT	AD1
2014-01-07	2014	1	7	AA	1892	0	2020-01-01	2020	1	1	6	1	GN
2014-02-14	2014	2	14	BB	182009	0	2020-01-01	2020	1	1	6	1	J
2014-11-20	2014	11	20	BB	82431	0	2020-01-01	2020	1	1	6	1	SC
2014-12-04	2014	12	4	BB	725	2	2020-01-01	2020	1	1	6	1	MP
2014-12-07	2014	12	7	AA	220691	2	2020-01-01	2020	1	1	6	1	JRR

	USER_ID	PAY_DAY	STORE_ID	WEEKDAY	GOODS_TYPE	COUNT	DATE
0	2858	1	1892	6	0	1	2020-01-01
1	5647	1	182009	6	0	1	2020-01-01
2	33314	1	82431	6	0	1	2020-01-01
3	37001	1	725	6	2	1	2020-01-01
4	37819	1	220691	6	2	1	2020-01-01

3. 모델링 구축 및 검증 - 2) Label Encoding 후 성능측정

Train Test set split 후 결과 측정

```
# 정해진 가이드 기준 Training, Test 데이터 분리
training_v1 = raw_data_v1.query("DATE >= '2020-01-01' and DATE <= '2020-09-30'")
test_v1 = raw_data_v1.query("DATE >= '2020-10-01' and DATE <= '2020-12-31'")

# Training, Test 데이터 준비
# USER_ID 외에 요일, 일, GOODS_TYPE, STORE_ID 적용
x_train = training_v1[["USER_ID", "PAY_DAY", "STORE_ID",
                       "WEEKDAY",  # 요일 관련 Column
                       "GOODS_TYPE"]]  # GOODS_TYPE 관련 Column
x_test = test_v1[["USER_ID", "PAY_DAY", "STORE_ID",
                       "WEEKDAY",  # 요일 관련 Column
                       "GOODS_TYPE"]]  # GOODS_TYPE 관련 Column
y_train = training_v1[["COUNT"]]
y_test = test_v1[["COUNT"]]
```

```
# 랜덤포레스트 모델 선언
RF = RandomForestRegressor()

# 랜덤포레스트 모델 학습
RF.fit(x_train, y_train)

# Test 데이터에 대한 예측 수행
predicted = RF.predict(x_test)

# MSE 및 MAE 측정
MSE = mean_squared_error(y_test, predicted)
MAE = mean_absolute_error(y_test, predicted)
```

```
MSE  : 0.05445964257607843
MAE  : 0.083495616109099
```

-> LabelEncoding 적용 후 예측 결과, MSE는 0.0545 수준, MAE는 0.0835수준으로 측정됨 확인

3. 모델링 구축 및 검증 - 2) OnehotEncoding 후 성능측정

Onehotencoding후 필요한 Feature만 추출

	USER_ID	PAY_DAY	STORE_ID	WEEKDAY_Mon	WEEKDAY_Tue	WEEKDAY_Wed	WEEKDAY_Thu	WEEKDAY_Fri	WEEKDAY_Sat	WEEKDAY_Sun	G
0	2858	1	1892	0	0	1	0	0	0	0	1
1	5647	1	182009	0	0	1	0	0	0	0	1
2	33314	1	82431	0	0	1	0	0	0	0	1
3	37001	1	725	0	0	1	0	0	0	0	0
4	37819	1	220691	0	0	1	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...
879266	1830551	31	219886	0	0	0	1	0	0	0	0
879267	1830570	31	82433	0	0	0	1	0	0	0	0
879268	1830580	31	92020	0	0	0	1	0	0	0	0
879269	1830589	31	92437	0	0	0	1	0	0	0	0
879270	1830598	31	220959	0	0	0	1	0	0	0	0

	USER_ID	PAY_DAY	STORE_ID	WEEKDAY_Mon	WEEKDAY_Tue	WEEKDAY_Wed	WEEKDAY_Thu	WEEKDAY_Fri	WEEKDAY_Sat	WEEKDAY_Sun	GC
0	2858	1	1892	0	0	1	0	0	0	0	1
1	5647	1	182009	0	0	1	0	0	0	0	1
2	33314	1	82431	0	0	1	0	0	0	0	1
3	37001	1	725	0	0	1	0	0	0	0	0
4	37819	1	220691	0	0	1	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...
623574	1709914	30	221022	0	0	1	0	0	0	0	0
623575	1709935	30	725	0	0	1	0	0	0	0	0
623576	1709942	30	223058	0	0	1	0	0	0	0	0
623577	1709950	30	182320	0	0	1	0	0	0	0	1
623578	1709952	30	105033	0	0	1	0	0	0	0	1

3. 모델링 구축 및 검증 - 2) OnehotEncoding 후 성능측정

Train Test set split 후 결과 측정

```
# 정해진 가이드 기준 Training, Test 데이터 분리
training_v1 = raw_data.query("DATE >= '2020-01-01' and DATE <= '2020-09-30'")
test_v1 = raw_data.query("DATE >= '2020-10-01' and DATE <= '2020-12-31'")

# Training, Test 데이터 준비
# USER_ID 외에 요일, 일, GOODS_TYPE, STORE_ID 적용
x_train = training_v1[["USER_ID", "PAY_DAY", "STORE_ID",
                       "WEEKDAY_Mon", "WEEKDAY_Tue", "WEEKDAY_Wed", "WEEKDAY_Thu", "WEEKDAY_Fri", "WEEKDAY_Sat", "WEEKI
                           "GOODS_TYPE_A", "GOODS_TYPE_B", "GOODS_TYPE_C", "GOODS_TYPE_D",]]  # GOODS_TYPE 관련 Column
x_test = test_v1[["USER_ID", "PAY_DAY", "STORE_ID",
                       "WEEKDAY_Mon", "WEEKDAY_Tue", "WEEKDAY_Wed", "WEEKDAY_Thu", "WEEKDAY_Fri", "WEEKDAY_Sat", "WEEKI
                           "GOODS_TYPE_A", "GOODS_TYPE_B", "GOODS_TYPE_C", "GOODS_TYPE_D",]]  # GOODS_TYPE 관련 Column
y_train = training_v1[["COUNT"]]
y_test = test_v1[["COUNT"]]
```

```
# 랜덤포레스트 모델 선언
RF = RandomForestRegressor()

# 랜덤포레스트 모델 학습
RF.fit(x_train, y_train)

# Test 데이터에 대한 예측 수행
predicted = RF.predict(x_test)

# MSE 및 MAE 측정
MSE = mean_squared_error(y_test, predicted)
MAE = mean_absolute_error(y_test, predicted)
```

```
MSE  : 0.05644967512275858
MAE  : 0.09009239898951775
```

- OneHotEncoding & Scaler & 가중치 적용 후 예측 결과, MSE는 0.0564, MAE는 0.0900 수준으로 측정됨 확인

4. 마무리

4. 마무리 - 정리

1. 가입자수는 꾸준히 증가
2. 결제한 년도 전체를 기준으로 결제한 달을 살펴 보면 11월이 가장 많았고, 금요일에 결제가 가장 많이 이루어졌다.
3. D_TYPE은 AA 타입이 56.09%(493,166건)으로 가장 많았다. 해당 데이터의 정보가 없어 모두의 주차장 어플, 모두의 주차장과 관련된 뉴스, 서울시 자료 등 조사해보았지만 정보가 나오지 않았다. 하지만 User ID와 연관이 있는 것으로 파악되어 회원 등급이지 않을까 추측해본다.
4. GOODS TYPE을 살펴 보니 A 타입이 앞도적으로 가장 많았다. 컬럼의 명을 비춰보아 왠지 결제시 사용하는 이용권(평일주중권, 평일야간권, 공휴일주중권, 공휴일야간권) 혹은 주차장의 종류(부설, 민영, 공영, 공유)로 추측해본다.
5. 주차를 가장 많이 이용한 지역은 J(중구) -> GN(강남) -> JRR(종로) 순이다.

유의미한 결론

1. 결제 날짜별 요일별 데이터와 GOODS_TYPE(주차권으로 추정)에 따라 데이터가 유의미해 보인다.
2. 결제 요일별 주차권은 각 주차권의 타입마다 다르다. A는 주말에 사용이 높고, B는금요일의 사용이 높다.
3. 결제 날짜 중 월별 데이터 마다 주차권 이용의 추이가 다르다 토요일 이용량이 많은 달은 2월 5월 10월, 주 초반 이용량이 많은 달은 3월 6월 9월, 금요일 이용량이 많은 달은 1월 7월 이다.

4. 마무리 - 추후 개선점

1. 정확한 모델링 구축에 실패 -> Feature를 다시 추려 모델링 정확도 올리기

2. 정확한 모델링 구축을 위해 -> 외부데이터인 교통량 데이터와 기상 데이터에서 필요한 데이터 추출, 우리가 지정한 Featrue와 merge하여 정확도 올리기

3. 정확한 모델링 구축을 위해 -> 결제일, 요일마다 이용 패턴이 다른 것으로 파악됨, 모델을 사용했을 때 GOODS_TYPE과 요일만 사용했지만 공휴일이라는 변수를 대입하여 모델 설정해보기

4. 예측 모델 사용시 랜덤 포레스트 모델 말고 선형회귀 모델, 분류 모델 등 다양한 모델을 사용해보기

참고 자료 출처

https://www.moduparking.com/ - 모두의 주차장 홈페이지

https://opengov.seoul.go.kr/sanction/16122885 - 모두의 주차장 공유사업 실행 계획

https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=we_are_youth&logNo=220828658137 - 모두의 주차장 플렛폼 소개

https://aboutstartup.tistory.com/3 - 기업소개

https://www.slideshare.net/cckslide/ss-54416200 - 민관협력을 통한 주차 공유 보고서

감사합니다