중고차 가격 예측

김민석 서지완 이영송 이유경 염예진

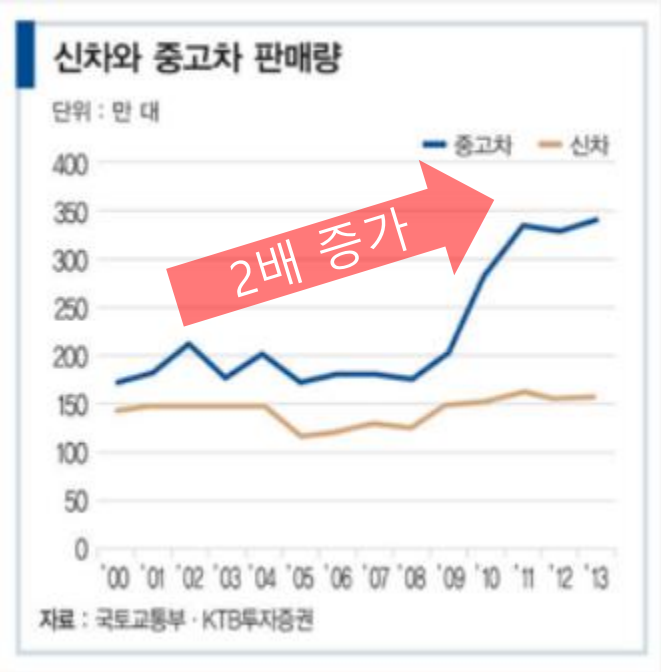
INDEX

01 프로젝트 개요

- 선정 배경 및 목적
- 프로젝트 타임라인

01 | 프로젝트 개요

- 선정 배경 및 목적



중고차 시장 성장세 BUT 판매자와 구매자간 정보 비대칭

>> 신뢰도 상승 필요!!

01 | 프로젝트 개요

- 프로젝트 타임라인

- 10일간 프로젝트 진행(2020.02.07 ~ 2020.02.16)

2/7	2/8	2/9	2/10	2/11	2/12	2/13	2/14	2/15	2/16	2/17
EDA	데이터 전처리 분석용 데이터셋 생성					모델 생성 변수 선택 모델 비교			SHAP	발표

02 데이터 파악 및 전처리

- EDA
- 연속형 변수 전처리
- 범주형 변수 전처리
- 상관관계 및 PCA
- 분석용 데이터셋 생성

02 EDA 및 전처리

- EDA

- Origin data : head(10)

	name	location	year	kilometers_driven	fuel_type	transmission	owner_type	mileage	engine	power	seats	new_price	price
0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5.0	NaN	1.75
1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0	NaN	12.50
2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5.0	8.61 Lakh	4.50
3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.0	NaN	6.00
4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5.0	NaN	17.74
5	Hyundai EON LPG Era Plus Option	Hyderabad	2012	75000	LPG	Manual	First	21.1 km/kg	814 CC	55.2 bhp	5.0	NaN	2.35
6	Nissan Micra Diesel XV	Jaipur	2013	86999	Diesel	Manual	First	23.08 kmpl	1461 CC	63.1 bhp	5.0	NaN	3.50
7	Toyota Innova Crysta 2.8 GX AT 8S	Mumbai	2016	36000	Diesel	Automatic	First	11.36 kmpl	2755 CC	171.5 bhp	8.0	21 Lakh	17.50
8	Volkswagen Vento Diesel Comfortline	Pune	2013	64430	Diesel	Manual	First	20.54 kmpl	1598 CC	103.6 bhp	5.0	NaN	5.20
9	Tata Indica Vista Quadrajet LS	Chennai	2012	65932	Diesel	Manual	Second	22.3 kmpl	1248 CC	74 bhp	5.0	NaN	1.95

Shape : (6919, 13)

02 EDA 및 전처리

- EDA

- 변수 형태 확인

변수명	TYPE	특징
Price	연속형	Target, 중고차 가격
Name	범주형	브랜드명 + 차종 (ex. Audi A4 New 2.0 TDI Multitronic)
Location	범주형	중고차를 구매한 도시 (11가지)
Year	연속형	중고차를 구매한 연도, max(Year)=2019
Kilometers_Driven	연속형	총 주행거리
Fuel_Type	범주형	연료 타입 (Petril, Diesel, Electric, CNG, LPG)
Transmission	범주형	변속기 유형 (Automatic, Manual)
Owner_Type	범주형	차량 소유주 타입 (first, second, third, fourth & above)
Mileage	범주형	표준 주행거리 (단위 kmpl 또는 km/kg)
Engine	범주형	엔진 (단위 CC)
Power	범주형	엔진의 최대 출력 (단위 bhp)
Seats	연속형	차량의 시트 수 (1~10)
New_Price	범주형	해당 차량의 새 차 가격 (화폐의 단위가 Cr, Lakh로 다르며 결측치가 많음)

- 결측치 개수 확인

```
name                  0
location              0
year                  0
kilometers_driven     0
fuel_type             0
transmission          0
owner_type            0
mileage               2
engine               36
power                36
seats                42
new_price          5195
price                 0
```

02 EDA 및 전처리

- EDA

- Name = brand_name + car_name

	name	location	year	kilometers_driven	fuel_type	transmission	owner_type	mileage	engine	power	seats	new_price	price	brand_name	car_name
0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5.0	NaN	1.75	Maruti	Wagon
1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0	NaN	12.50	Hyundai	Creta
2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5.0	8.61 Lakh	4.50	Honda	Jazz
3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.0	NaN	6.00	Maruti	Ertiga
4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5.0	NaN	17.74	Audi	A4

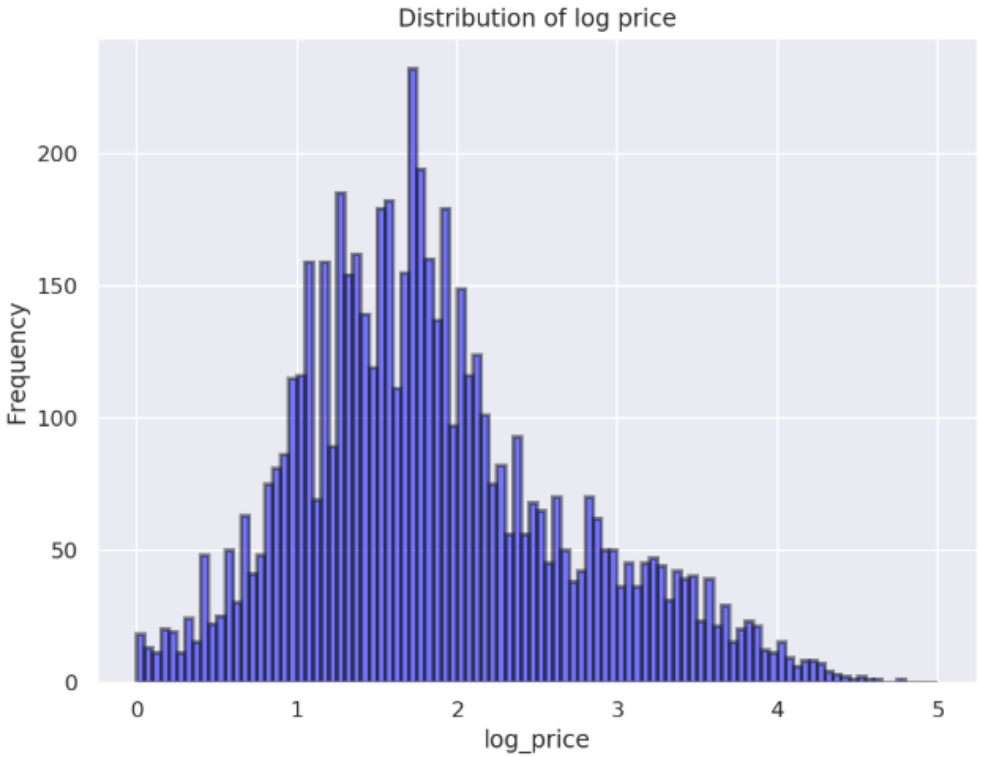
1) Name의 띄어쓰기 기준 첫 번째 단어 : brand_name
2) Name의 띄어쓰기 기준 두 번째 단어 : car_name

02 EDA 및 전처리

- EDA

- price 분포 확인

- log(price) 분포 확인

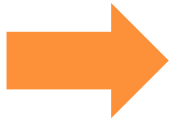


02 EDA 및 전처리

- 연속형 변수 전처리

- 연속형 변수 : **Year**, Kilometer_Driven, Mileage, Engine, Power, Seats, New_Price, Price

year
2010
2015
2011
2012
2013



year
10
5
9
8
7

2019년까지의 데이터가 존재한다.
연식을 나타내기 위한 변수로 변환

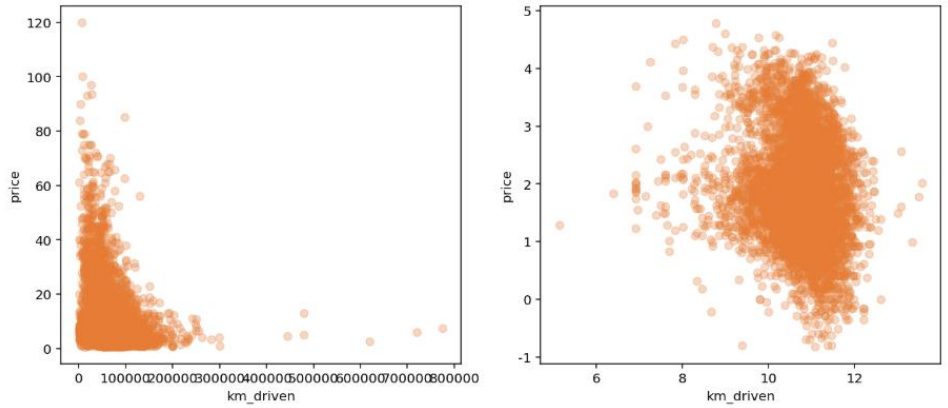
>> 2019 – YEAR + 1

02 EDA 및 전처리

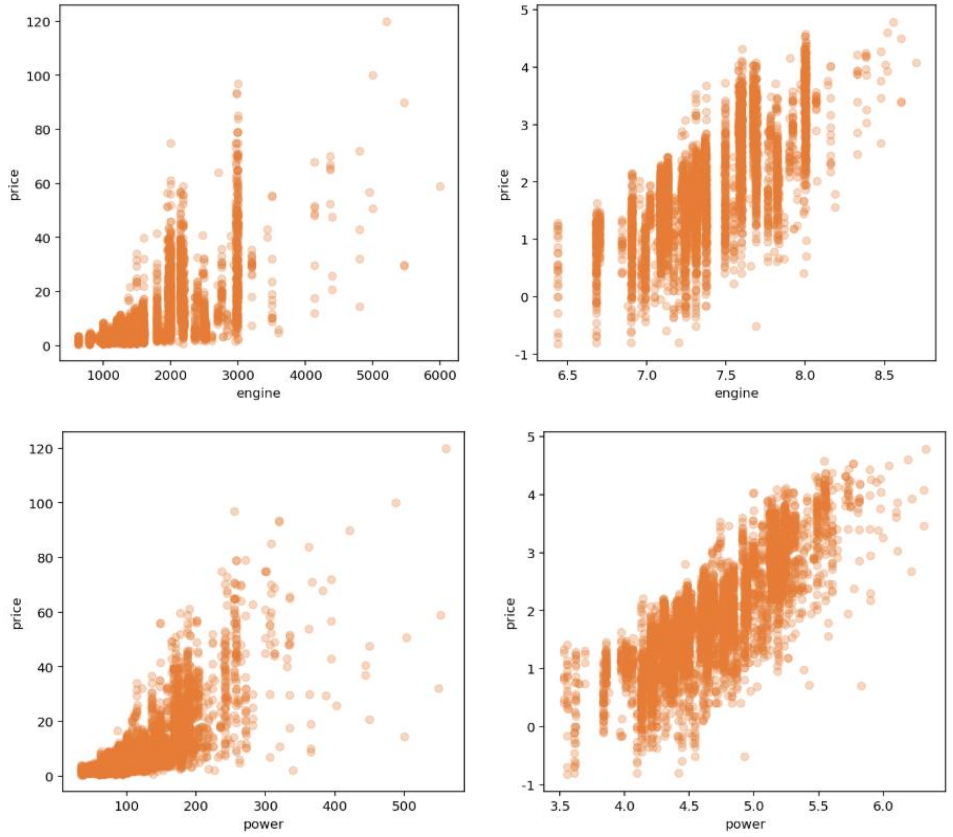
- 연속형 변수 전처리

- 연속형 변수 : Year, **Kilometer_Driven, Mileage, Engine, Power**, Seats, New_Price, Price

Log scale X **Log scale O**



Log scale X **Log scale O**

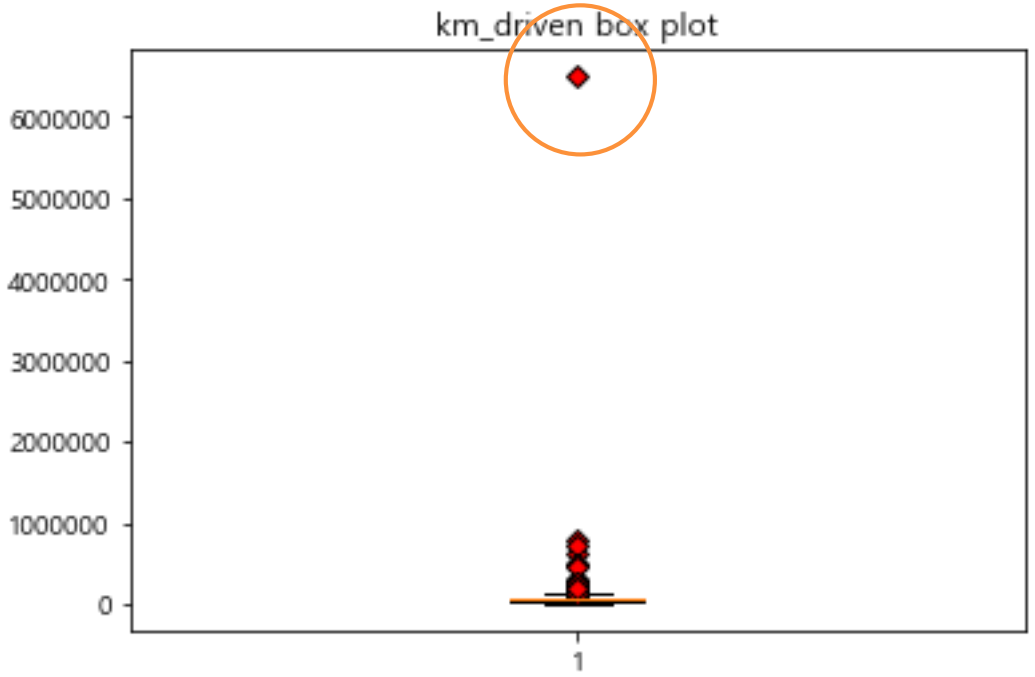


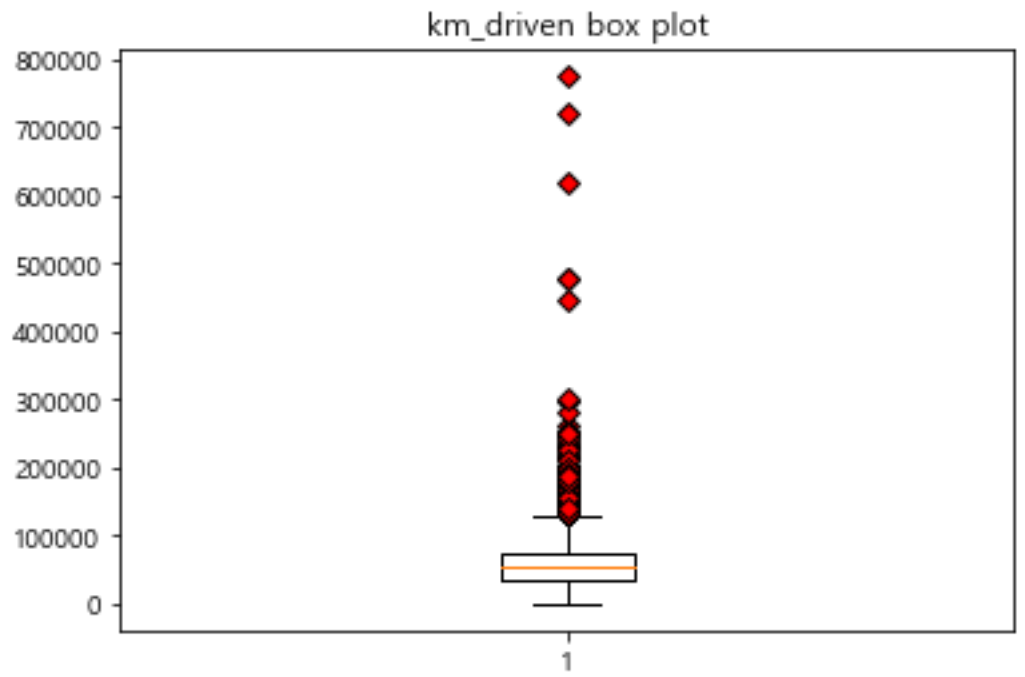
>> Log scaling한 것이 price(target)와의 분포가 퍼져있으므로 **y와 연속형 변수에 log scaling**을 한 변수를 사용

02 EDA 및 전처리

- 연속형 변수 전처리

- 연속형 변수 : Year, **Kilometer_Driven**, Mileage, Engine, Power, Seats, New_Price, Price





6,500,000 이상치를 발견했지만, 차종이 같은 데이터를 확인한 결과 오탈자로 판단함

>> 6500000 / 1000

02 EDA 및 전처리

- 연속형 변수 전처리

- 연속형 변수 : Year, Kilometer_Driven, **Mileage**, Engine, Power, Seats, New_Price, Price

	name	location	year	km_driven	fuel_type	transmission	owner_type	mileage	engine	power	seats	new_price	price	brand_name	car_name
4446	Mahindra E Verito D4	Chennai	4	50000	Electric	Automatic	First	NaN	72 CC	41 bhp	5.0	13.58 Lakh	13.00	Mahindra	E
4904	Toyota Prius 2009-2016 Z4	Mumbai	9	44000	Electric	Automatic	First	NaN	1798 CC	73 bhp	5.0	NaN	12.75	Toyota	Prius

- 결측치 2개 발견, fuel_type 이 Electric 인 것을 확인함 **>> Fuel_type 중 Electric 이 총 2개 뿐이기 때문에, 삭제함**

	name	location	year	km_driven	fuel_type	transmission	owner_type	mileage	engine	power	seats	new_price	price	brand_name	car_name
14	Land Rover Freelander 2 TD4 SE	Pune	8	85000	Diesel	Automatic	Second	0.0	2179.0	115.00	5.0	NaN	17.50	Land Rover	Rover
67	Mercedes-Benz C-Class Progressive C 220d	Coimbatore	1	15369	Diesel	Automatic	First	0.0	1950.0	194.00	5.0	49.14 Lakh	35.67	Mercedes-Benz	C-Class
79	Hyundai Santro Xing XL	Hyderabad	15	87591	Petrol	Manual	First	0.0	1086.0	NaN	5.0	NaN	1.30	Hyundai	Santro
194	Honda City 1.5 GXI	Ahmedabad	13	60006	Petrol	Manual	First	0.0	NaN	NaN	NaN	NaN	2.95	Honda	City
229	Ford Figo Diesel	Bangalore	5	70436	Diesel	Manual	First	0.0	1498.0	99.00	NaN	NaN	3.60	Ford	Figo
262	Hyundai Santro Xing XL	Hyderabad	14	99000	Petrol	Manual	First	0.0	1086.0	NaN	5.0	NaN	1.75	Hyundai	Santro
307	Hyundai Santro Xing XL	Chennai	14	58000	Petrol	Manual	Second	0.0	1086.0	NaN	5.0	NaN	1.50	Hyundai	Santro

- 0 값 대체 : 해당 car_name의 power 평균값으로 대체, 채워지지 않은 값은 brand_name의 power 평균값으로 대체

02 EDA 및 전처리

- 연속형 변수 전처리

- 연속형 변수 : Year, Kilometer_Driven, Mileage, **Engine**, Power, Seats, New_Price, Price

	name	location	year	km_driven	fuel_type	transmission	owner_type	mileage	engine	power	seats	new_price	price	brand_name	car_name
194	Honda City 1.5 GXI	Ahmedabad	13	60006	Petrol	Manual	First	18.22	NaN	NaN	NaN	NaN	2.95	Honda	City
208	Maruti Swift 1.3 VXi	Kolkata	10	42001	Petrol	Manual	First	16.10	NaN	NaN	NaN	NaN	2.11	Maruti	Swift
733	Maruti Swift 1.3 VXi	Chennai	14	97800	Petrol	Manual	Third	16.10	NaN	NaN	NaN	NaN	1.75	Maruti	Swift
749	Land Rover Range Rover 3.0 D	Mumbai	12	55001	Diesel	Automatic	Second	10.96	NaN	NaN	NaN	NaN	26.50	Land Rover	Rover
1294	Honda City 1.3 DX	Delhi	11	55005	Petrol	Manual	First	12.80	NaN	NaN	NaN	NaN	3.20	Honda	City
1327	Maruti Swift 1.3 ZXI	Hyderabad	5	50295	Petrol	Manual	First	16.10	NaN	NaN	NaN	NaN	5.80	Maruti	Swift
1385	Honda City 1.5 GXI	Pune	16	115000	Petrol	Manual	Second	18.22	NaN	NaN	NaN	NaN	1.50	Honda	City

Car_name의 engine평균값

```
City : 1494.72
Swift : 1240.47
Rover : 2475.72
Santro : 1077.42
Etios : 1373.57
5 : 2359.43
Wagon : 1014.45
CR-V : 2216.69
Punto : 1172.0
Jazz : 1272.33
```

	name	location	year	km_driven	fuel_type	transmission	owner_type	mileage	engine	power	seats	new_price	price	brand_name	car_name
0	Maruti Wagon R LXI CNG	Mumbai	10	72000	CNG	Manual	First	26.60	998.0	58.16	5.0	NaN	1.75	Maruti	Wagon
1	Hyundai Creta 1.6 CRDi SX Option	Pune	5	41000	Diesel	Manual	First	19.67	1582.0	126.20	5.0	NaN	12.50	Hyundai	Creta
2	Honda Jazz V	Chennai	9	46000	Petrol	Manual	First	18.20	1199.0	88.70	5.0	8.61 Lakh	4.50	Honda	Jazz
3	Maruti Ertiga VDI	Chennai	8	87000	Diesel	Manual	First	20.77	1248.0	88.76	7.0	NaN	6.00	Maruti	Ertiga
4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	7	40670	Diesel	Automatic	Second	15.20	1968.0	140.80	5.0	NaN	17.74	Audi	A4

1) Engine의 단위 제거 (CC)
2) 결측치가 속한 car_name의 평균값으로 대체

02 | EDA 및 전처리

- 연속형 변수 전처리

- 연속형 변수 : Year, Kilometer_Driven, Mileage, Engine, **Power**, Seats, New_Price, Price

	name	location	year	km_driven	fuel_type	transmission	owner_type	mileage	engine	power	seats	new_price	price	brand_name	car_name
76	Ford Fiesta 1.4 SXi TDCi	Jaipur	12	111111	Diesel	Manual	First	17.80	1399.00	NaN	5.0	NaN	2.00	Ford	Fiesta
79	Hyundai Santro Xing XL	Hyderabad	15	87591	Petrol	Manual	First	10.14	1086.00	NaN	5.0	NaN	1.30	Hyundai	Santro
89	Hyundai Santro Xing XO	Hyderabad	13	73745	Petrol	Manual	First	17.00	1086.00	NaN	5.0	NaN	2.10	Hyundai	Santro
120	Hyundai Santro Xing XL eRLX Euro III	Mumbai	15	102000	Petrol	Manual	Second	17.00	1086.00	NaN	5.0	NaN	0.85	Hyundai	Santro
143	Hyundai Santro Xing XO eRLX Euro II	Kochi	12	80759	Petrol	Manual	Third	17.00	1086.00	NaN	5.0	NaN	1.67	Hyundai	Santro

1) Power의 단위 제거 (bhp)
2) 0, 결측치 대체 : 해당 car_name의 power 평균값으로 대체

```
Fiesta : 74.27
Santro : 62.28
City : 110.78
Swift : 77.88
Jetta : 125.79
Indica : 67.68
Rover : 189.6
Etios : 76.91
Fortwo : nan
Cayman : nan
Petra : nan
Baleno : 83.08
Optra : 112.41
Bolero : 63.05
Micra : 68.61
E-Class : 204.43
Jeep : nan
Qualis : 75.0
Estilo : nan
5 : 209.25
Wagon : 66.03
Teana : 179.5
CR-V : 153.77
Esteem : 85.0
Endeavour : 169.18
A4 : 163.63
Punto : 67.0
Jazz : 91.25
Siena : nan
```

02 EDA 및 전처리

- 연속형 변수 전처리

- 연속형 변수 : Year, Kilometer_Driven, Mileage, Engine, Power, **Seats**, New_Price, Price

	name	location	year	km_driven	fuel_type	transmission	owner_type	mileage	engine	power	seats	new_price	price	brand_name	car_name
194	Honda City 1.5 GXI	Ahmedabad	13	60006	Petrol	Manual	First	18.22	1494.72	110.78	NaN	NaN	2.95	Honda	City
208	Maruti Swift 1.3 VXi	Kolkata	10	42001	Petrol	Manual	First	16.10	1240.47	77.38	NaN	NaN	2.11	Maruti	Swift
229	Ford Figo Diesel	Bangalore	5	70436	Diesel	Manual	First	19.44	1498.00	99.00	NaN	NaN	3.60	Ford	Figo
733	Maruti Swift 1.3 VXi	Chennai	14	97800	Petrol	Manual	Third	16.10	1240.47	77.38	NaN	NaN	1.75	Maruti	Swift
749	Land Rover Range Rover 3.0 D	Mumbai	12	55001	Diesel	Automatic	Second	10.96	2475.72	189.60	NaN	NaN	26.50	Land Rover	Rover

- 0, 결측치 처리 : 동일한 name의 평균값으로 대체하려고 함
- But, 동일한 name이 존재하지 않아서 car_name의 평균값으로 대체

```
City : 5.0
Swift : 5.0
Figo : 5.0
Rover : 7.0
Santro : 5.0
Etios : 5.0
5 : 5.0
Wagon : 5.0
Endeavour : 7.0
CR-V : 5.0
Punto : 5.0
Jazz : 5.0
Estilo : 5.0
```

02 EDA 및 전처리

- 연속형 변수 전처리

- 연속형 변수 : Year, Kilometer_Driven, Mileage, Engine, Power, Seats, **New_Price,** Price

name	location	year	km_driven	fuel_type	transmission	owner_type	mileage	engine	power	seats	new_price	price	brand_name	car_name	yn_new_price
Maruti Wagon R LXI CNG	Mumbai	10	72000	CNG	Manual	First	26.60	998.0	58.16	5.0	NaN	1.75	Maruti	Wagon	0
Hyundai Creta 1.6 CRDi SX Option	Pune	5	41000	Diesel	Manual	First	19.67	1582.0	126.20	5.0	NaN	12.50	Hyundai	Creta	0
Honda Jazz V	Chennai	9	46000	Petrol	Manual	First	18.20	1199.0	88.70	5.0	8.61 Lakh	4.50	Honda	Jazz	1
Maruti Ertiga VDI	Chennai	8	87000	Diesel	Manual	First	20.77	1248.0	88.76	7.0	NaN	6.00	Maruti	Ertiga	0
Audi A4 New 2.0 TDI Multitronic	Coimbatore	7	40670	Diesel	Automatic	Second	15.20	1968.0	140.80	5.0	NaN	17.74	Audi	A4	0

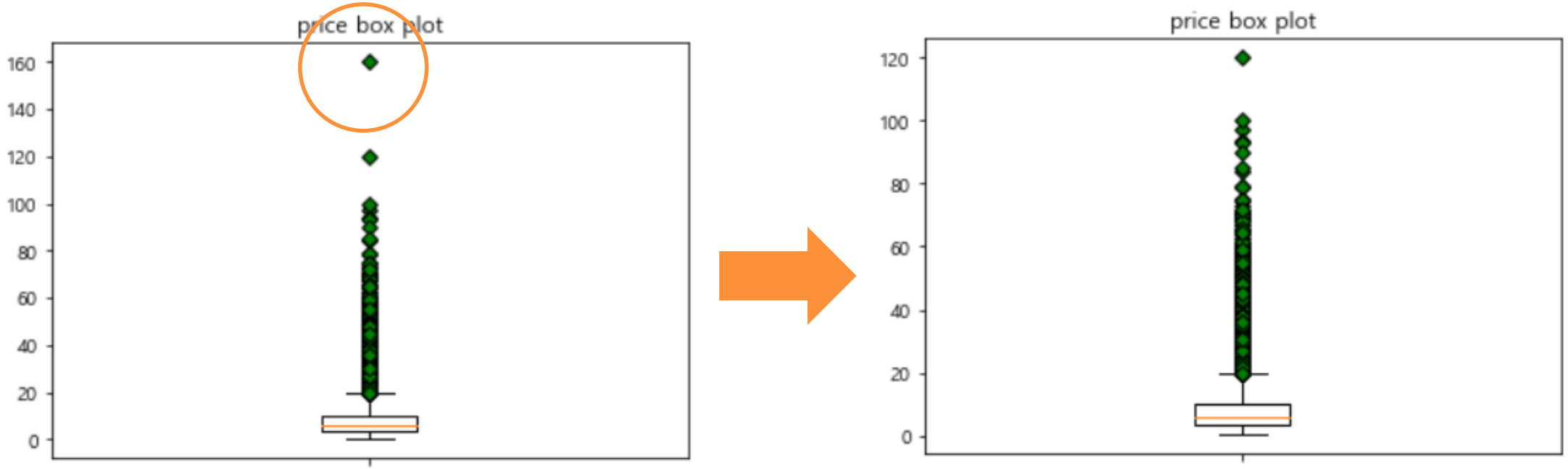
- New_price의 결측치 : 약 70% 보유
  -> 해당 변수값은 삭제
  -> 대신, 파생변수 yn_new_price 생성

- New_price의 값이 존재하면, yn_new_price=1
- New_price의 값이 존재하지 않으면, yn_new_price=0

02 | EDA 및 전처리

- 연속형 변수 전처리

- 연속형 변수 : Year, Kilometer_Driven, Mileage, Engine, Power, Seats, New_Price, **Price**

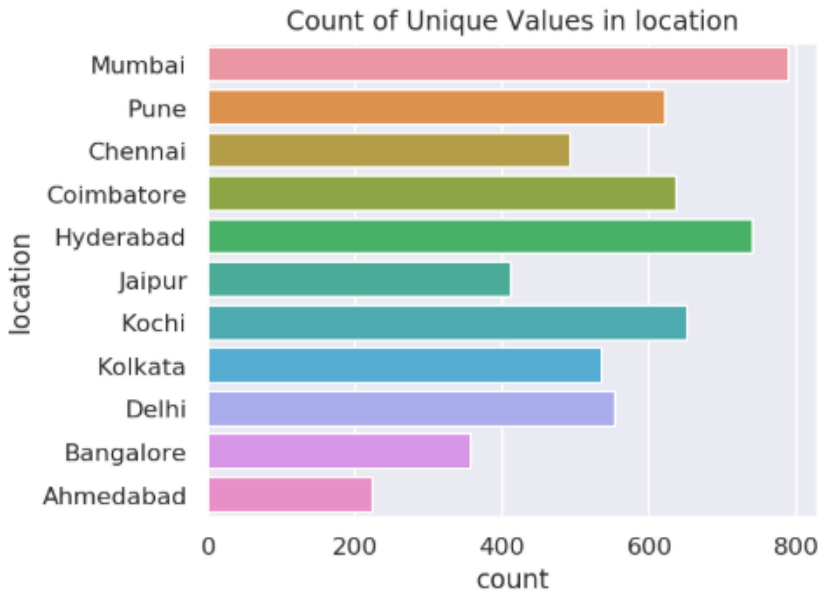


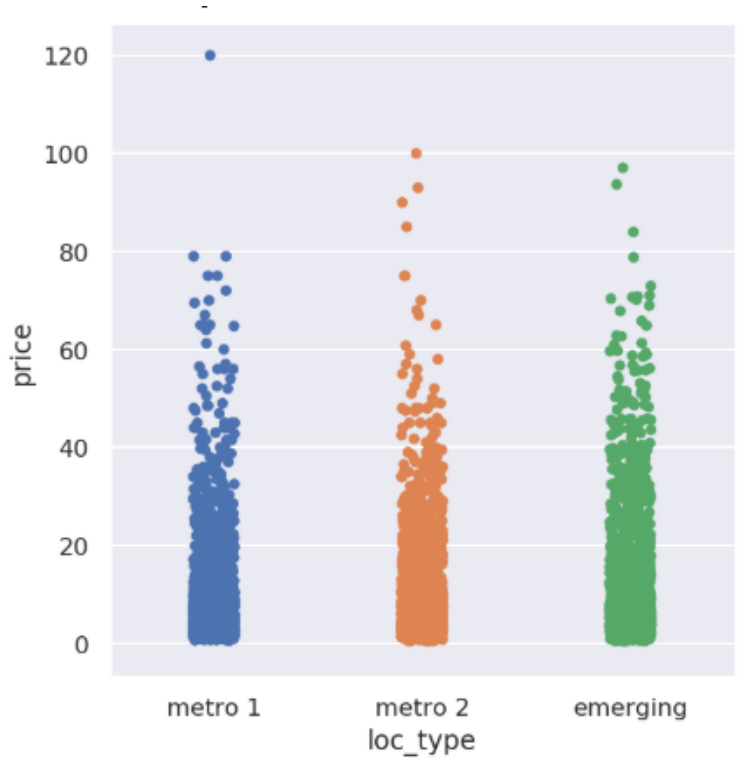
- Price가 160이상인 것의 name = Land Rover Range Rover 3.0 Diesel LWB Vogue
  >> 이상치라고 판단, 제거

02 | EDA 및 전처리

- 범수형 변수 전처리

- 범주형 변수 : **Location**, Fuel_type, transmission, owner_type, brand_name



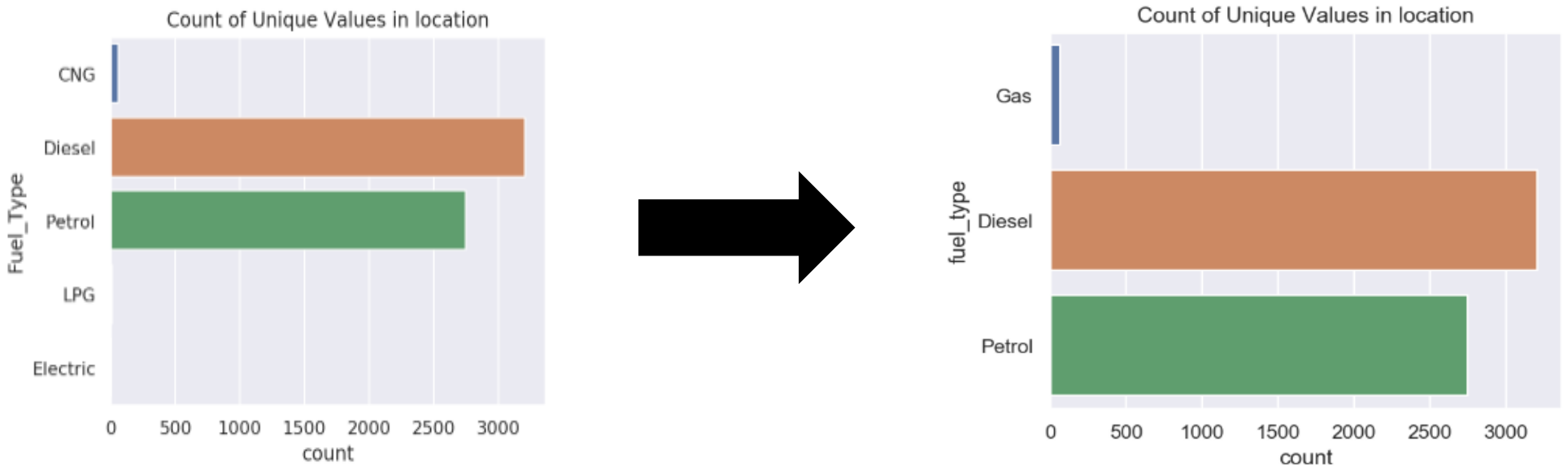


- 인도 도시 특성에 맞게 범주를 다시 구분함
  - mubai/deli/col/chenai -> metro 1
  - bangalroo/ hyd/ameda/pune -> metro 2
  - zaipuru / cochi/ coinbatro -> metro 3

02 EDA 및 전처리

- 범수형 변수 전처리

- 범주형 변수 : Location, **Fuel_type**, transmission, owner_type, brand_name

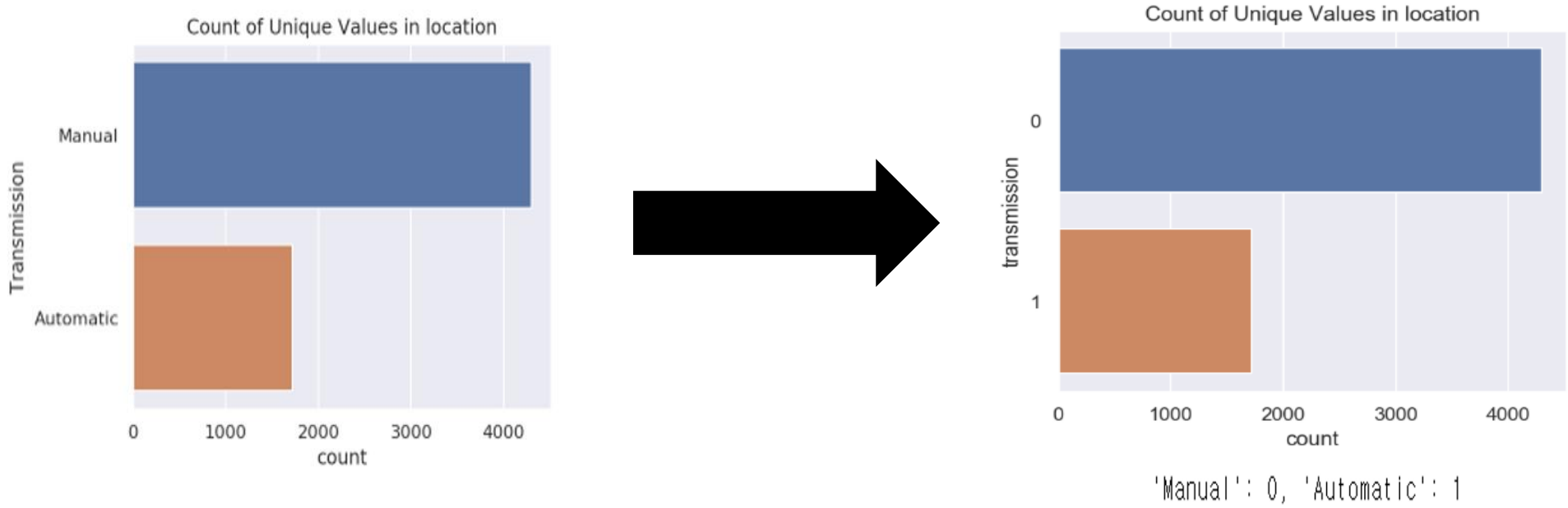


- Diesel, Petrol, Gas(CNG, LPG)로 다시 범주화함
- Electric은 2개뿐이라 삭제

02 | EDA 및 전처리

- 범수형 변수 전처리

- 범주형 변수 : Location, Fuel_type, **transmission,** owner_type, brand_name

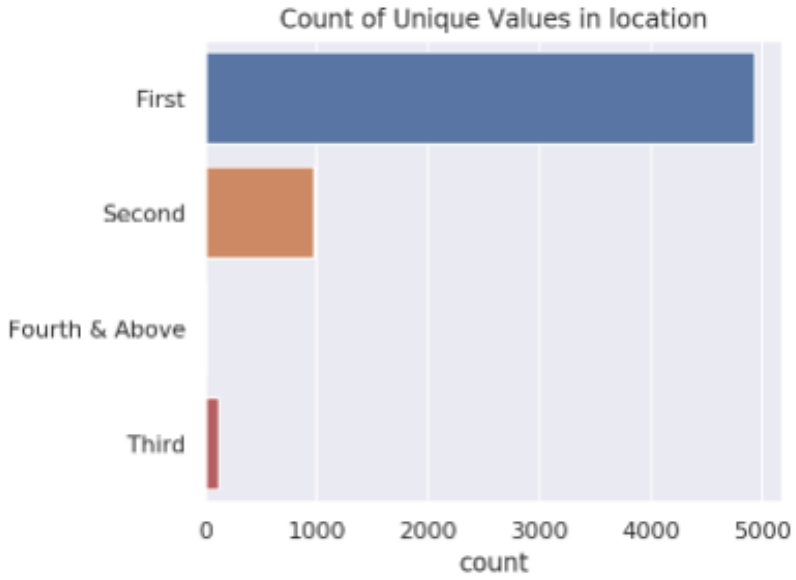


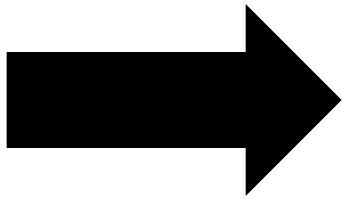
- Manual : 0, Automatic : 1 로 변환

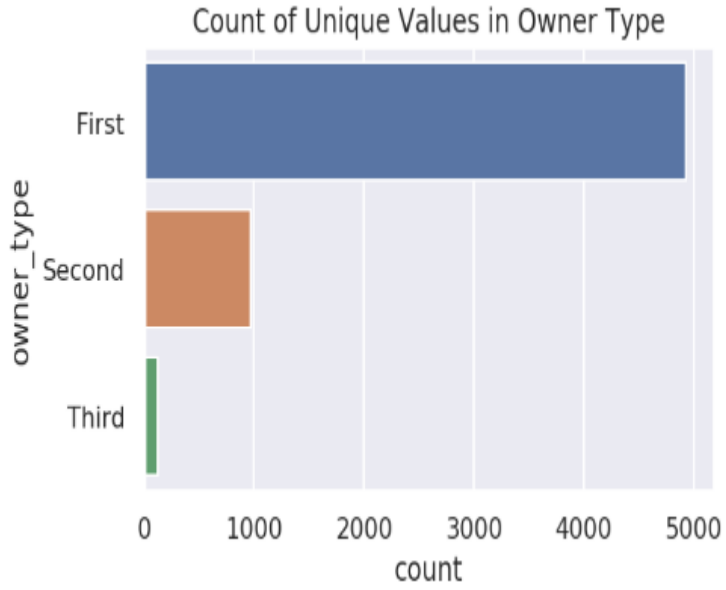
02 | EDA 및 전처리

- 범수형 변수 전처리

- 범주형 변수 : Location, Fuel_type, transmission, **owner_type**, brand_name







- First, Second, Third, Fourth & Above를 다시 범주화함
- First, Second, Third(Third + Fourth & Above)

02 EDA 및 전처리

- 범수형 변수 전처리

- 범주형 변수 : Location, Fuel_type, transmission, owner_type, **brand_name**

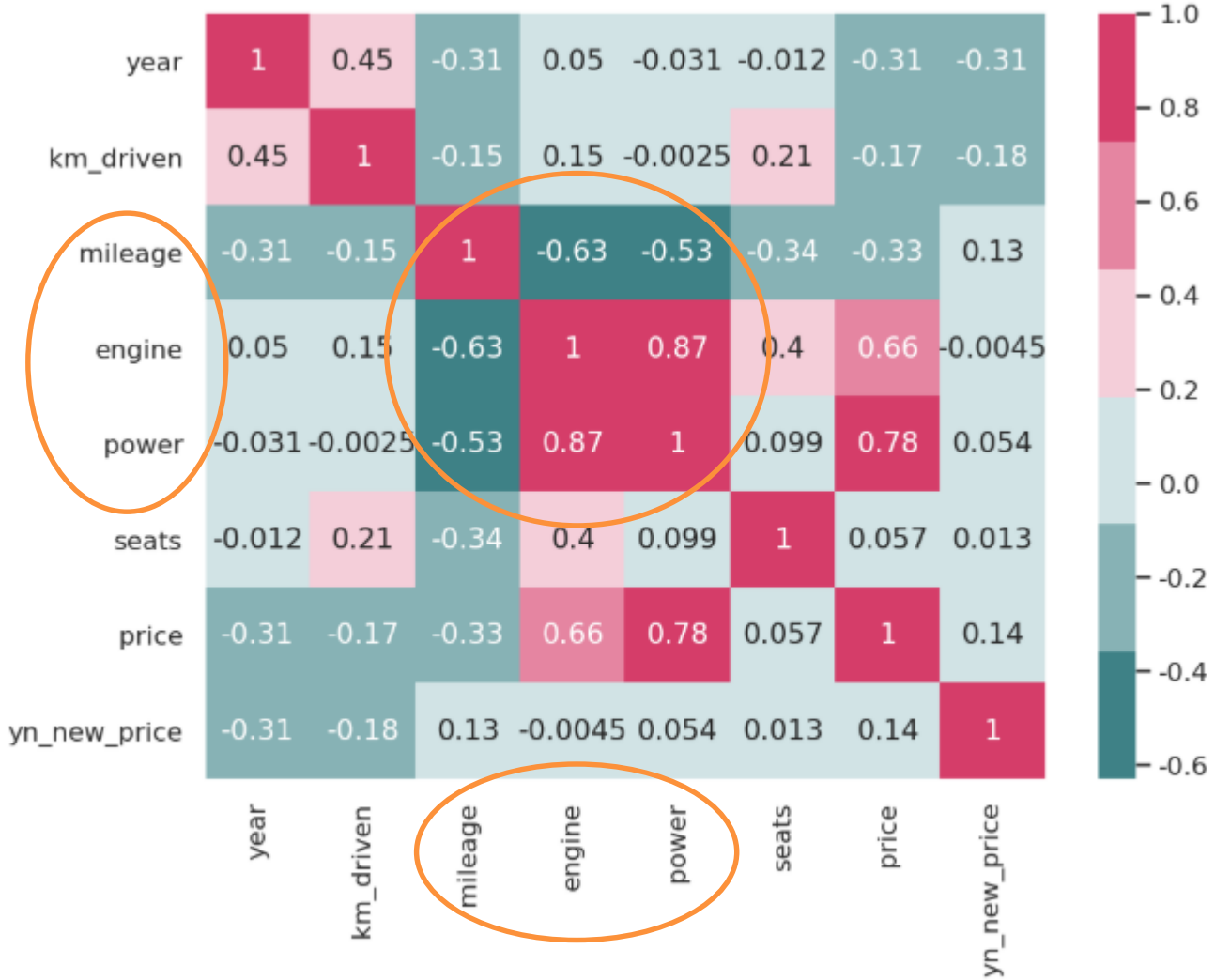
Volvo	21	0.349011
Porsche	18	0.299152
Jeep	15	0.249294
Datsun	13	0.216055
sparse_brand	10	0.166196

	count	ratio
Force One	3	0.049867
Isuzu	3	0.049867
Lamborghini	1	0.016622
Bentley	1	0.016622
Ambassador	1	0.016622
Smart Fortwo	1	0.016622

- Brand_name의 개수가 10개 이하인 경우 sparse_brand로 분류
- Sparse_brand = (Force One, Isuzu, Lamborghini, Bentley, Ambassador, Smart Fortwo)

02 EDA 및 전처리

- 상관관계 및 PCA



①

- Mileage, Engine 상관관계 = -0.63
- Mileage, Power 상관관계 = -0.53
- Engine, Power 상관관계 = 0.87

=> PCA를 통해 3가지 변수를
하나의 벡터로 표현하기

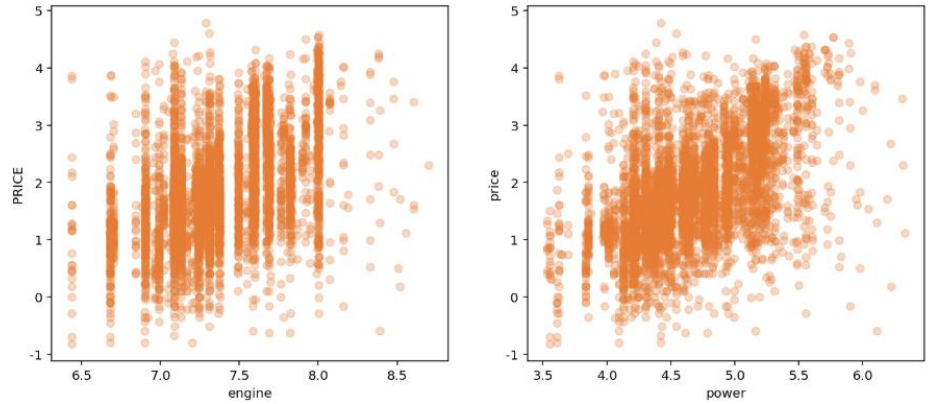
②

- Mileage, Engine, Power
  PC1의 정보량 = 0.74
  버려지는 정보량이 크다고 판단

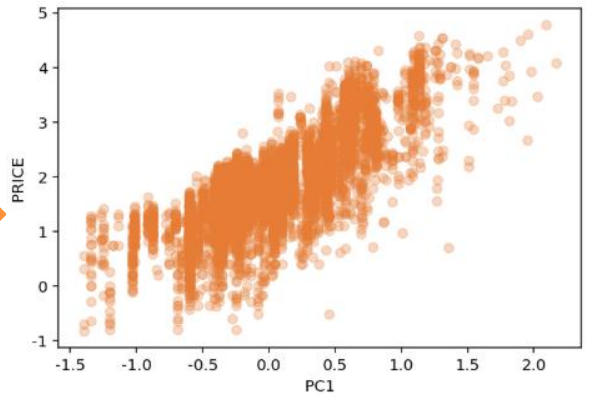
>> 상관관계가 더 높은 Engine과 Power
두 가지 변수로만 PCA하기로 결정

02 EDA 및 전처리

- 상관관계 및 PCA



- Power와 Price, Engine과 Price의 scatter plot




```
PC1 정보량 :  0.81
singular vector :
 [[ 0.61976547  0.78478708]
 [ 0.78478708 -0.61976547]]
```

- Power와 Price, Engine과 Price의 scatter plot

➢ 두 가지 변수를 사용하는 것이 아니라 정보량이 0.81인 PC1을 사용하기로 결정

02 EDA 및 전처리

- 변수 선택

RFECV BY RF (3-Fold)

	feature	rank
0	engine	1
27	brand_name_Honda	1
28	brand_name_Hyundai	1
29	brand_name_Jaguar	1
30	brand_name_Jeep	1
31	brand_name_Land Rover	1
32	brand_name_Mahindra	1
33	brand_name_Maruti	1
34	brand_name_Mercedes-Benz	1
35	brand_name_Mini	1
26	brand_name_Ford	1
36	brand_name_Mitsubishi	1
40	brand_name_Skoda	1
41	brand_name_Tata	1
42	brand_name_Toyota	1
43	brand_name_Volkswagen	1
44	brand_name_Volvo	1
45	brand_name_sparse_brand	1
46	yn_new_price_0	1
47	yn_new_price_1	1
48	loc_type_emerging	1
38	brand_name_Porsche	1
49	loc_type_metro 1	1
50	loc_type_metro 2	1
13	seats_2.0	1
3	km_driven	1
7	fuel_type_Petrol	1
8	transmission_Automatic	1
9	transmission_Manual	1
10	owner_type_First	1
23	brand_name_Chevrolet	1
11	owner_type_Second	1
12	owner_type_Third	1
5	fuel_type_Diesel	1
4	year	1
15	seats_5.0	1
16	seats_6.0	1
17	seats_7.0	1
2	mileage	1
1	power	1
21	brand_name_Audi	1
22	brand_name_BMW	1
14	seats_4.0	1
39	brand_name_Renault	2
37	brand_name_Nissan	3
18	seats_8.0	4
25	brand_name_Fiat	5
6	fuel_type_Gas	6
20	seats_10.0	7
24	brand_name_Datsun	8
19	seats_9.0	9

➢1 만 선택하여, 총 43개 선택

02 | EDA 및 전처리

- 최종 데이터셋 생성

1) **기본** 데이터셋 : log scaling **X** >> price + 연속형 변수+ 범주형 변수

2) **로그** 데이터셋 : PCA + log scaling **O** >> log(price) + log(연속형 변수) + 범주형 변수

03 모델 생성 및 예측

- Linear Regression
- Support Vector Machine
- Tree 기반 모델

03 모델 생성 및 예측

- LR / SVM / TREE 모델 예측 (5-Fold CV)

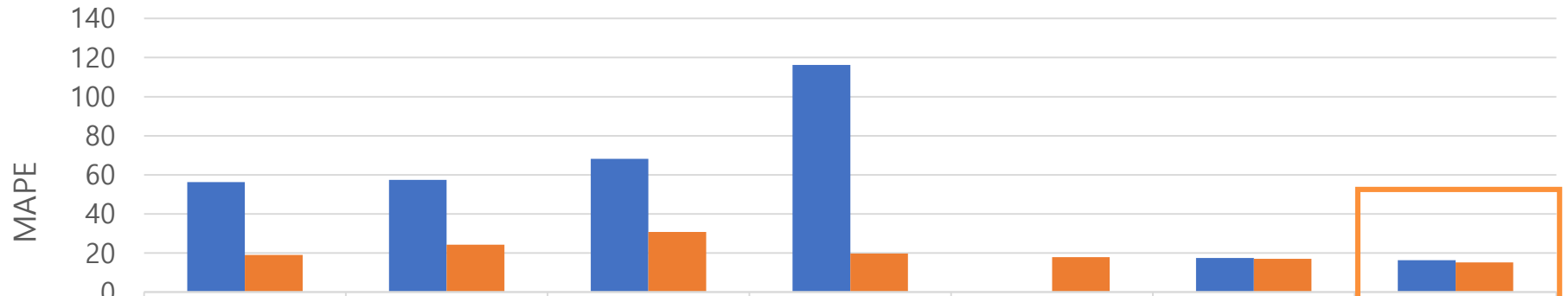
모델	LR				SVM		TREE	
	OLS	LASSO 람다:0.01	LASSO 람다:0.1	SGD(LSE)	Linear	KERNEL (poly:2)	RF	XGB
평가 기준	MAPE	MAPE	MAPE	MAPE	MAPE	MAPE	MAPE	MAPE
기본 데이터셋	56.208	57.434	68.215	적합X	116.25	적합X	17.466	16.330
로그 데이터셋	19.039	24.195	30.730	19.598	19.784	17.808	17.058	15.235

04 모델 평가 및 해석

- 모델별 비교 및 평가
- SHAP 사용 후 평가

04 모델 평가 및 해석

- LR / SVM / TREE 모델 예측 비교



	OLS	LASSO_0.01	LASSO_0.1	Linear SVR	Kernel Poly(2) SVR	RF	XGB
기본 데이터셋	56.208	57.434	68.215	116.25		17.466	16.33
로그 데이터셋	19.039	24.195	30.73	19.784	17.808	17.058	15.235

- 로그 변환이나 PCA등의 전처리 절차 후 선형 모델의 예측 정확도 증가
- 또한, 비선형모델이나 scale의 영향을 받지 않는 Tree모델에서도 약간의 정확도 향상을 볼 수 있었음

- 가장 높은 예측력을 보이는 모델은 **XGBoost** 정확한 **predict**가 필요한 목적인 경우, 이 모델을 사용하는 것이 적절

- 하지만, 일반 OLS모델을 사용하는 경우에도 비교적 높은 정확도가 보여짐
  -> **Inference**가 중요하다면 **OLS 모델**을 사용하는 것도 적합해 보임

04 | 모델 평가 및 해석

- SHAP 사용 후 평가

추가 과제 1. XGBoost에서도 Inference가 필요한 경우는 어떻게 하지?
2. 정확도를 더 높이기 위해서는 어떤 변수가 필요할까?

→ **SHAP !**

< 실제값 vs 예측값 비율 차이 TOP 7 >

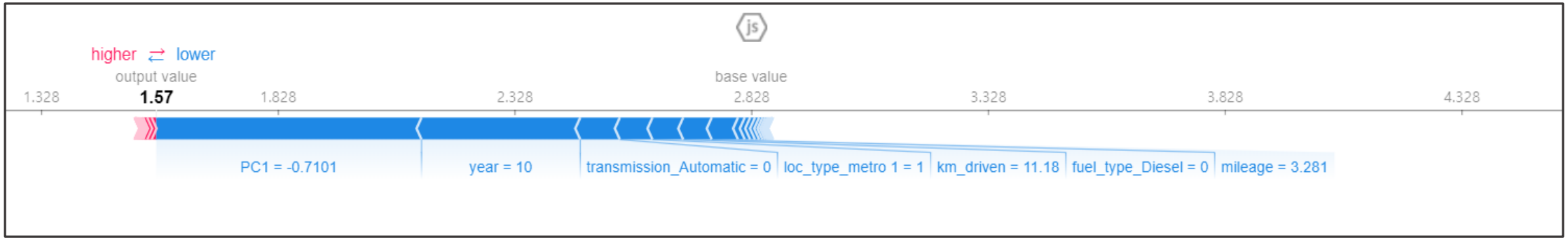
	real	pred
458	83.96	25.27
220	40.00	12.04
148	75.00	29.92
1108	1.52	0.65
1035	14.45	6.74
1041	1.60	0.79
1158	9.80	4.87

→ **SHAP**을 활용하여,
458번의 사례를 확인해보자 !

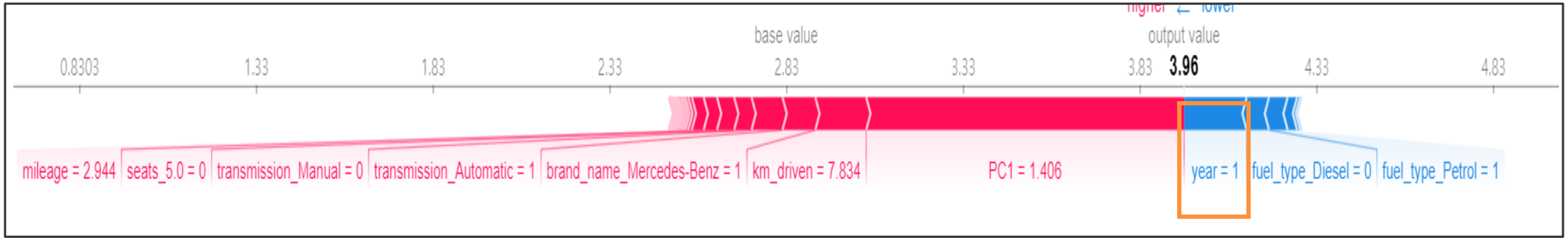
04 | 모델 평가 및 해석

- SHAP 사용 후 평가

Correct prediction



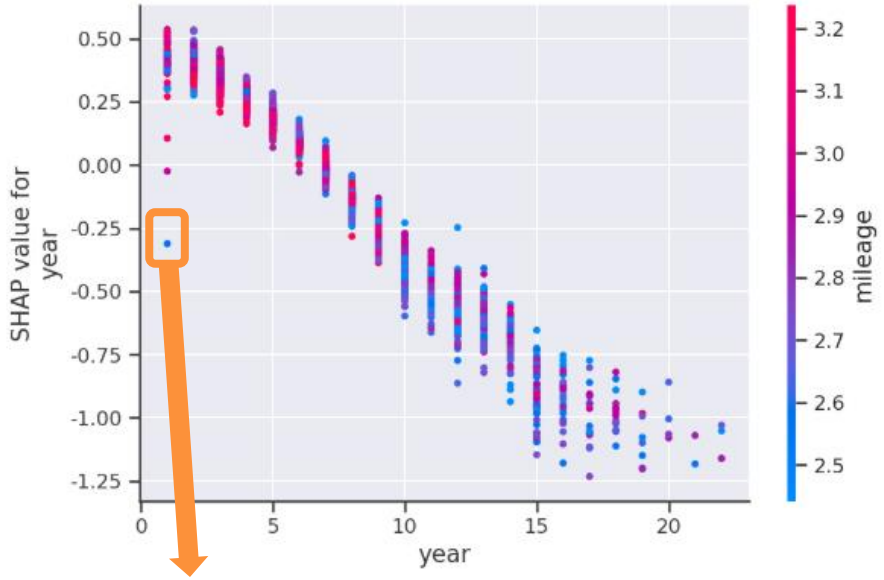
Wrong prediction (458번)



- 실제값 83.96 = np.exp(5.4 – 1) 이 아닌 25.27 = np.exp(3.96 - 1) 을 예측한 사례
- 다른 변수들은 일반적으로 적용되나, 1년 연식의 차임에도 predict 값을 lower 시키는 변수로써 적용함

04 모델 평가 및 해석

- SHAP 사용 후 평가



	name	location	year	...	power	seats	new_price	price	brand_name	car_name
	Mercedes-Benz SLC 43 AMG	Coimbatore	1	...	362.07	2.0	1.06 Cr	83.96	Mercedes-Benz	SLC

- 1년 된 차라는 사실은 다른 case들에 대해서는 양(+)의 영향을 주나 해당 case는 음(-)의 영향을 주는 **outlier**라 판단
- 실제 값을 관찰한 결과 SLC라는 차종, 즉 다른 스펙 (ex. Seats 개수) 이 비슷하더라도 차 가격이 비싼 **스포츠형 차**라는 사실을 잡아 주지 못했을 것으로 판단 -> 특수 차종을 알 수 있는 **차량유형 변수의 추가** 필요
- 전반적으로 신차 가격을 알 수 있는 변수 (New_Price)를 결측치가 많아 사용하지 못하였기에 더욱 정확한 예측이 어려웠던 것으로 판단 (p18)
  - > 크롤링 등으로 모든 모델의 대한 신규 차 가격을 파악 후, **변수 (New Price)** 의 활용 필요

THANK YOU