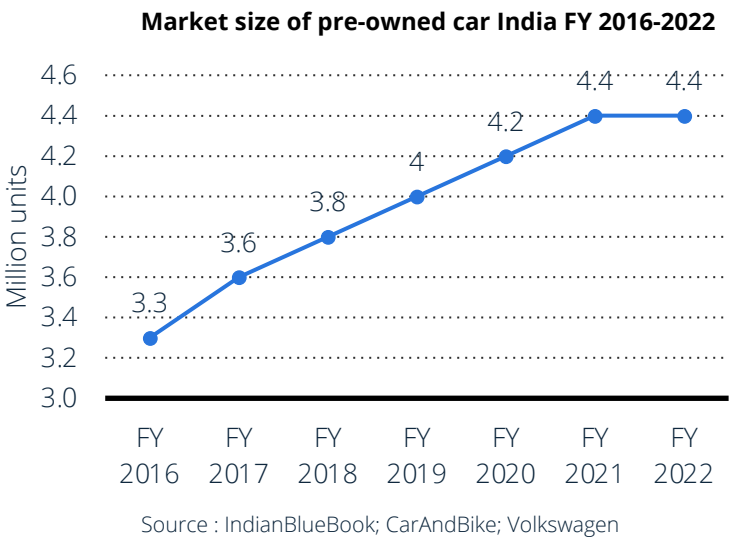
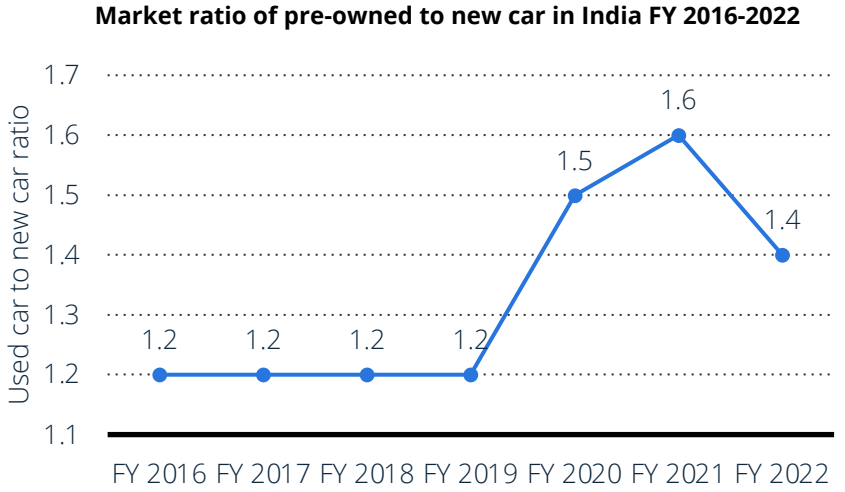
중고차 가격 예측 모델 및 개선 방안

A2반 손 민 수



Source : IndianBlueBook; CarAndBike; Volkswagen



Source : IndianBlueBook; CarAndBike

IndianBlueBook magazine에 따르면 최근 중고차 시장은 지속적인 성장세를 보이고 있으며, 이미 2016년 부터 신차 시장과 중고차 시장 비율이 1:1.2로 지속적으로 성장하고 있음을 알 수 있다.

따라서 가격과 공급이 불안정한 중고차 가격을 예측할 수 있다면, 고객들의 니즈 파악을 하여 꾸준한 공급을 유지시켜 수익실현을 할 수 있을 것이다.

따라서 중고차 가격을 효과적으로 예측할 수 있는 핵심 영향인자 도출과 가격 예측 모델을 개발하고자 한다.

분석 계획

1. 데이터 Type 별 이상치 및 결측치 확인 후 제거
    1. 연속형 자료 이상치 및 결측치 확인 후 제거
    2. 범주형 자료 이상치 및 결측치 확인 후 제거
2. 설명변수 간의 상관관계 확인
3. 목표변수 그래프 EDA
4. 데이터 모델
    1. 회귀분석 모델
    2. 의사결정나무 모델
    3. 그래디언트 부스팅 모델
5. 모델 주요 인자 비교 및 확인
6. 결론

데이터 현황

변수명	정상 데이터 수	데이터 유형
Name	7253	문자형
Location	7253	문자형
Price	6200	숫자형
Year	7253	숫자형
Kilometers_Driven	7253	숫자형
Fuel_Type	7253	문자형
Transmission	7253	문자형
Owner_Type	7253	문자형
Mileage	7251	문자형
Engine	7207	문자형
Power	7207	문자형
Seats	7200	숫자형
New_Price	1006	문자형

결측치를 가지고 있는 경우

변수명	결측치 수	데이터 유형
Price	1053	숫자형
Mileage	2	문자형
Engine	46	문자형
Power	46	문자형
Seats	53	숫자형
New_Price	6247	문자형

데이터의 유형이 잘못된 경우

변수명	데이터 유형(현재)	데이터 유형 (목표)
Mileage	문자형	숫자형
Engine	문자형	숫자형
Power	문자형	숫자형
New_Price	문자형	숫자형

데이터 정제 (데이터 단위 제거 및 타입 변환)

	Mileage	Engine	New_Price
0	26.6 kmpl	998 CC	NaN
1	19.67 kmpl	1582 CC	NaN
2	18.2 kmpl	1199 CC	8.61 Lakh
3	20.77 kmpl	1248 CC	NaN
4	15.2 kmpl	1968 CC	NaN
5	21.1 kmpl	814 CC	NaN
6	23.08 kmpl	1461 CC	NaN
7	11.36 kmpl	2755 CC	21 Lakh
8	20.54 kmpl	1598 CC	NaN
9	22.3 kmpl	1248 CC	NaN

	Power
0	58.16 bhp
1	126.2 bhp
2	88.7 bhp
3	88.76 bhp
4	140.8 bhp

	Power
76	null bhp
79	null bhp
89	null bhp
120	null bhp
143	null bhp

	Mileage	Engine	Power	New_Price
0	26.60	998.0	58.16	NaN
1	19.67	1582.0	126.20	NaN
2	18.20	1199.0	88.70	8.61
3	20.77	1248.0	88.76	NaN
4	15.20	1968.0	140.80	NaN
5	21.10	814.0	55.20	NaN
6	23.08	1461.0	63.10	NaN
7	11.36	2755.0	171.50	21.00
8	20.54	1598.0	103.60	NaN
9	22.30	1248.0	74.00	NaN

Mileage(kmpl), Engine(CC), New_Price(Lakh) 문자를 삭제한 뒤, 데이터를 float64 형태로 바꿔주었다.

Power는 bhp 단위가 있는 데이터 뿐만 아니라 null bhp 로 null 값에 단위가 붙어있는 데이터 가 존재하였다.

따라서 이를 제거해준 뒤, float64 형태로 바꾸어 주었다.

데이터 정제 (데이터 단위 환산)

	New_Price
0	NaN
1	NaN
2	8.61
3	NaN
4	NaN
5	NaN
6	NaN
7	21.00
8	NaN
9	NaN

	New_Price
0	NaN
1	NaN
2	13586.58
3	NaN
4	NaN
5	NaN
6	NaN
7	33138.00
8	NaN
9	NaN

1 루피 = 15.76원 (2023-08-08 기준)
1 Lakh = 100,000루피

New_Price 와 Price 간의 단위를 맞춰주기 위해서 천원 단위로 통일해 주도록 하겠다.

EDA 순서

1. 연속형 자료

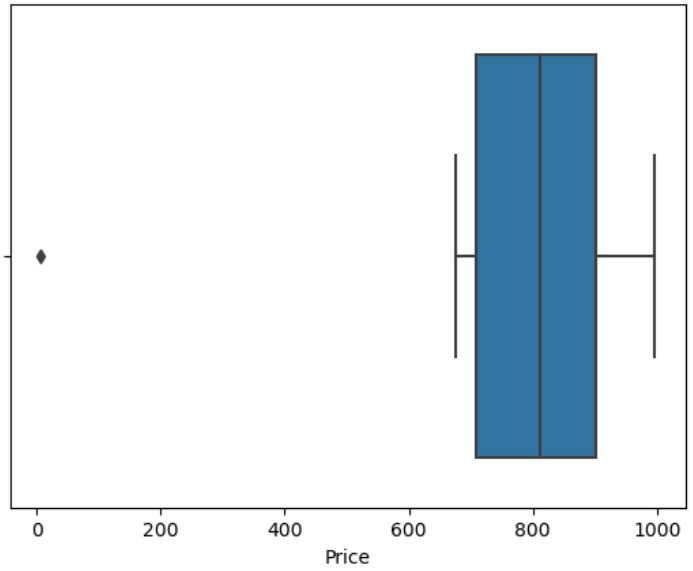
변수명	정상 데이터 수	데이터 유형
Price	6200	숫자형
Year	7253	숫자형
Kilometers_Driven	7253	문자형
Mileage	7251	숫자형
Engine	7207	숫자형
Power	7207	숫자형
Seats	7200	숫자형
New_Price	1006	숫자형

2. 문자형 자료

변수명	정상 데이터 수	데이터 유형
Name	7253	문자형
Location	7253	문자형
Fuel_Type	7253	문자형
Transmission	7253	문자형
Owner_Type	7253	문자형

1. Price – 이상치 제거

	Price
count	6200.000000
mean	14912.514750
std	17674.318464
min	7.080000
25%	5365.360000
50%	8814.520000
75%	15869.972500
max	245273.600000



Price 의 최소값이 7임을 알 수 있다.

하지만 7,000원 가격의 중고차가 존재할 수 없으므로 이는 이상치라고 판단하고 추가적으로 그래프를 그려보겠다.

1000 이하의 데이터에 대해서 그래프를 그려본 결과 7의 값을 가지고 있는 데이터만 존재한다.

따라서 7의 값의 행을 제거해 주도록 하겠다.

1. Price – 결측치 제거

	결측치 개수	결측치 비율
계	1053	0.145201

데이터의 결측치의 비율이 14%로 높다.
하지만 Price 는 목표 변수로 결측치를 채우면 모델에 큰 영향을 미칠 것으로 보인다. 따라서 데이터의 손실이 있어도 모델의 정확도를 높이기 위해 데이터를 지워주도록 하겠다.

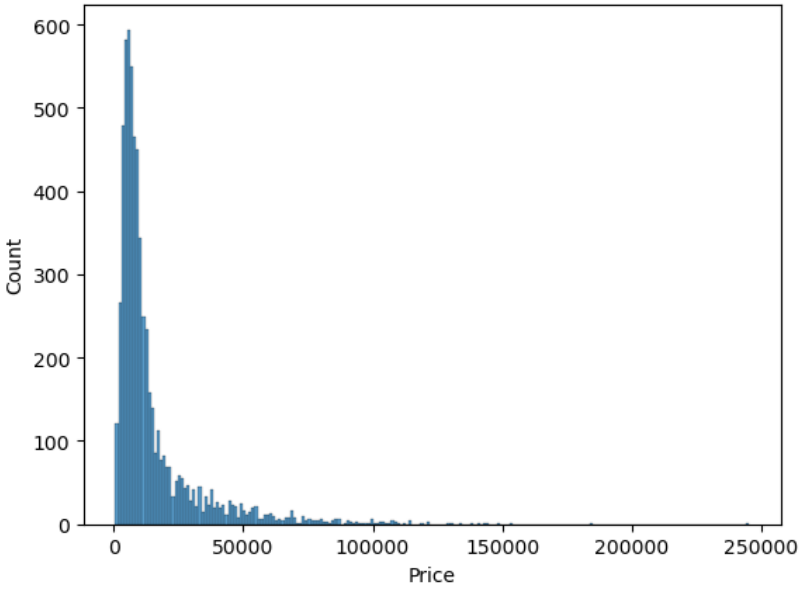
(7252, 13)

(6199, 13)

데이터의 행 개수가 7252 에서 6199 로 약 14%의 데이터 손실이 있다.

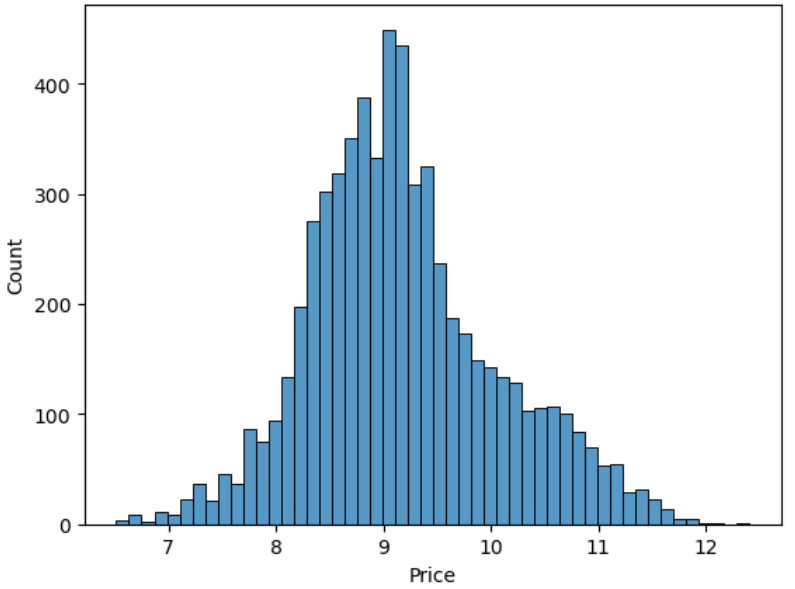
1. Price – log 변환

변환 전



데이터가 왼쪽으로 심하게 치우친 경향을 볼 수 있다.
따라서 이를 Log 변환을 통해 데이터의 정규성을 확보해 보도록 하겠다.

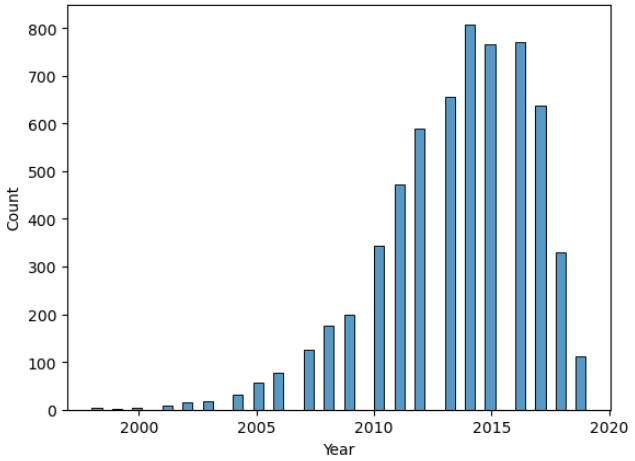
변환 후



데이터가 정규분포 모양을 어느정도 따르고 있다.

2. Year

	Year
count	6199.000000
mean	2013.433457
std	3.271719
min	1998.000000
25%	2012.000000
50%	2014.000000
75%	2016.000000
max	2019.000000

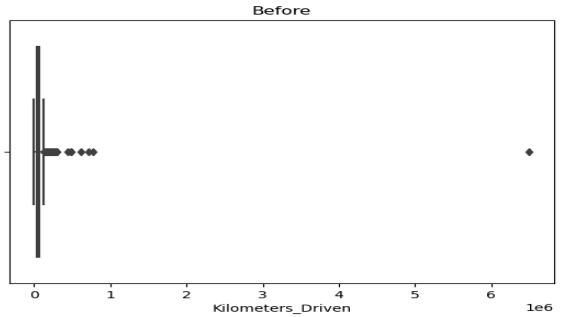


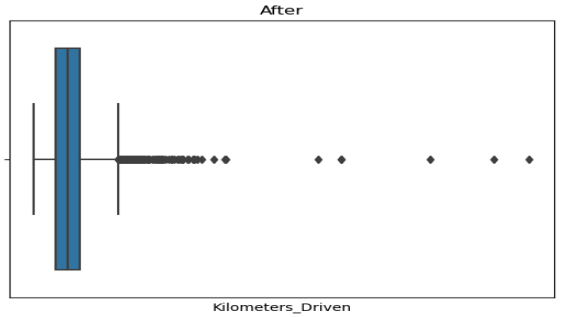
Year의 데이터에서는 1998년 부터 2019년 까지의 데이터가 있는 것으로 확인 되었고, 년식이 최신일수록 중고차 차량이 많아지는 것을 확인할 수 있었다.

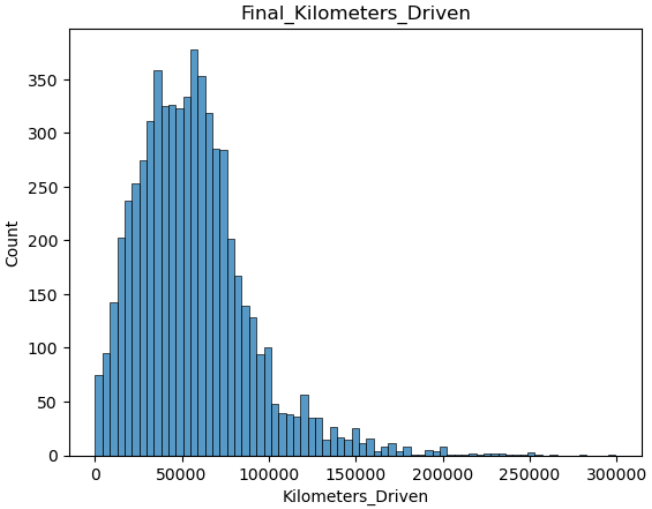
또한 연식이 최신으로 갈수록 Price 또한 점점 증가하는 것을 볼 수 있었다.
따라서 Year 변수는 Price를 예측하는데 영향을 미친다고 볼 수 있다.

3. Kilometers_Driven – 이상치 제거

	Kilometers_Driven
count	6199.00
mean	58162.90
std	90112.49
min	171.00
25%	33000.00
50%	52516.00
75%	72302.00
max	6500000.00







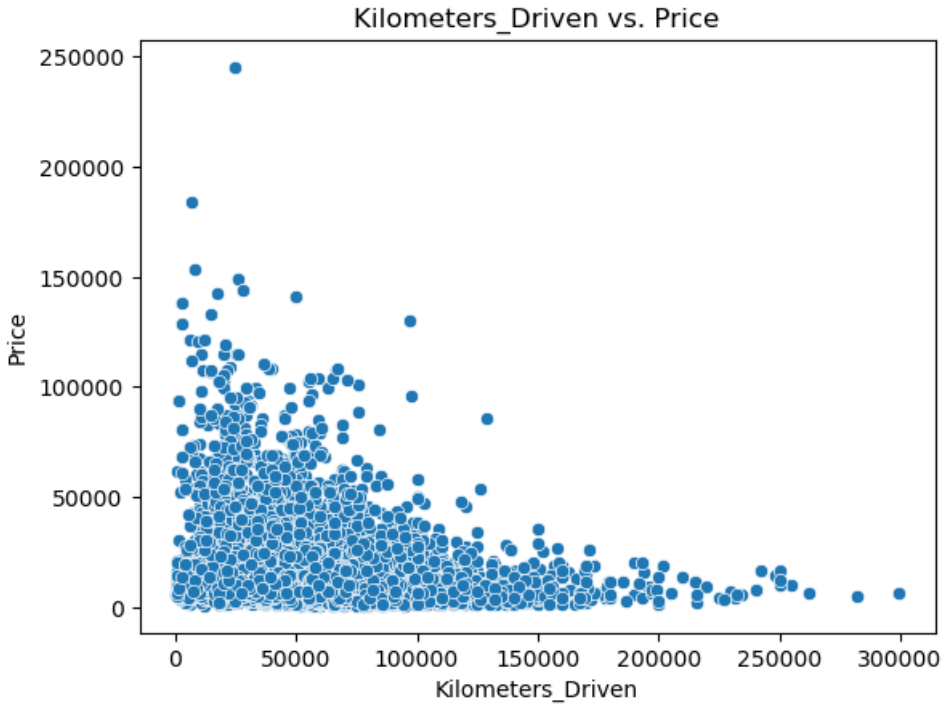
Kilometers_Driven 의 연속형 데이터의 특징과 boxplot을 그려본 결과 6,500,000 km 라는 상식과 맞지 않은 데이터가 존재하는 것을 확인하였다.

이는 확실한 이상치이므로 지워주도록 하겠다.

그 이후 데이터를 확인해 본 결과 아직도 많은 이상치가 존재하는 것을 확인하였고, 300,000km 이상 주행한 차량의 비율을 확인해본 결과 0.97%로 매우 낮은 개수의 차량이 존재하는 것을 확인하였다.
따라서 이또한 이상치로 판단하여 제거해 주었다.

제거 한 뒤, 그래프(Final_Kilometers_Driven)을 그려본 결과 그래프의 모양이 많이 나아진 것을 확인해 볼 수 있었다.

3. Kilometers_Driven – Price와의 비교

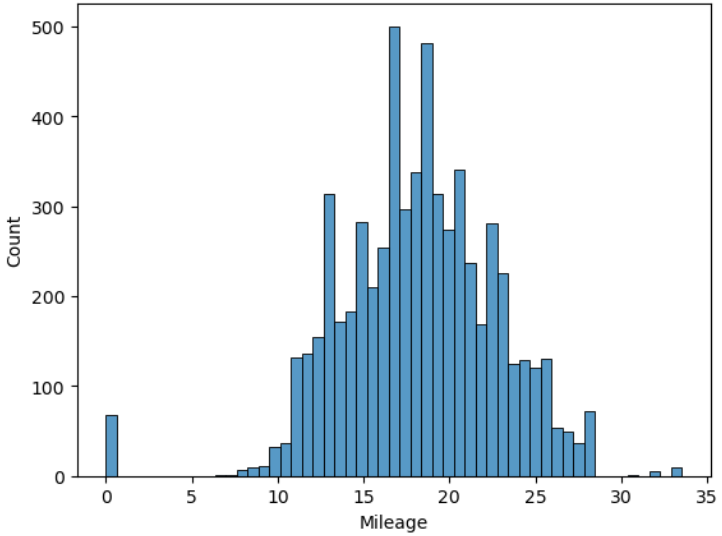


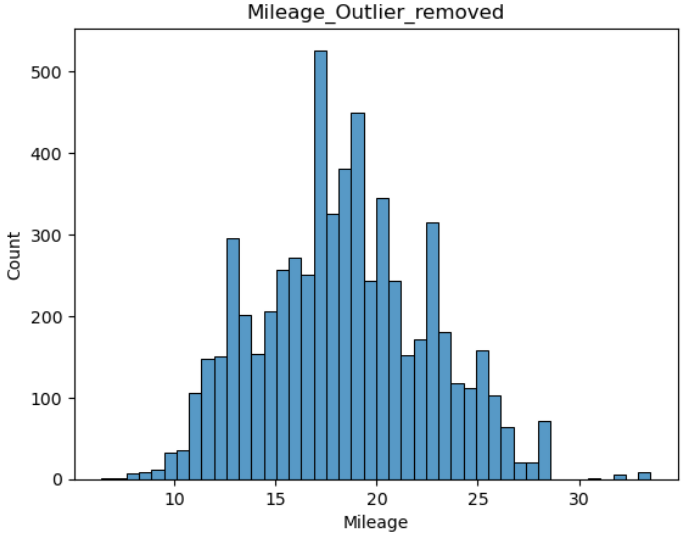
Price와 비교해본 결과 Kilometers_Driven이 적을 수록 Price 가 높게 분포되는 경향이 있다.

따라서 Kilometers_Driven 변수는 Price에 영향을 미친다고 할 수 있다.

4. Mileage – 이상치 제거

	Mileage
count	6189.000
mean	18.183
std	4.581
min	0.000
25%	15.260
50%	18.200
75%	21.100
max	33.540





데이터의 분포를 확인해본 결과 Mileage 가 0인 데이터가 존재하였다.

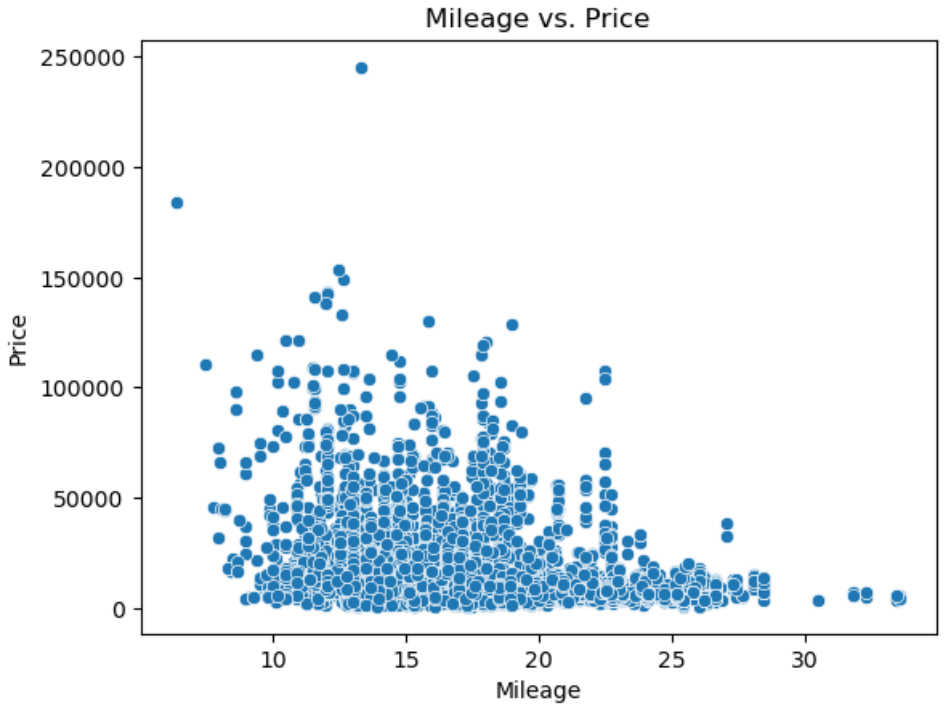
Mileage는 자동차 회사가 지정해준 값이므로, 데이터 내에서 자동차 명과 회사가 같을 경우 그 데이터를 가져와 데이터의 손실을 최대한 줄이고자 하였다.

또한 전기차 데이터는 Mileage 를 0으로 가지고 있었는데, 데이터가 2개 만 존재하여 지워주도록 하겠다.

그 결과 데이터 수를 2개 만 제거하여 손실율을 최소화 시켜 주었다.

또한 그래프가 이전 그래프보다 정규성을 더 만족하는 것으로 보인다.

4. Mileage – Price 와 비교

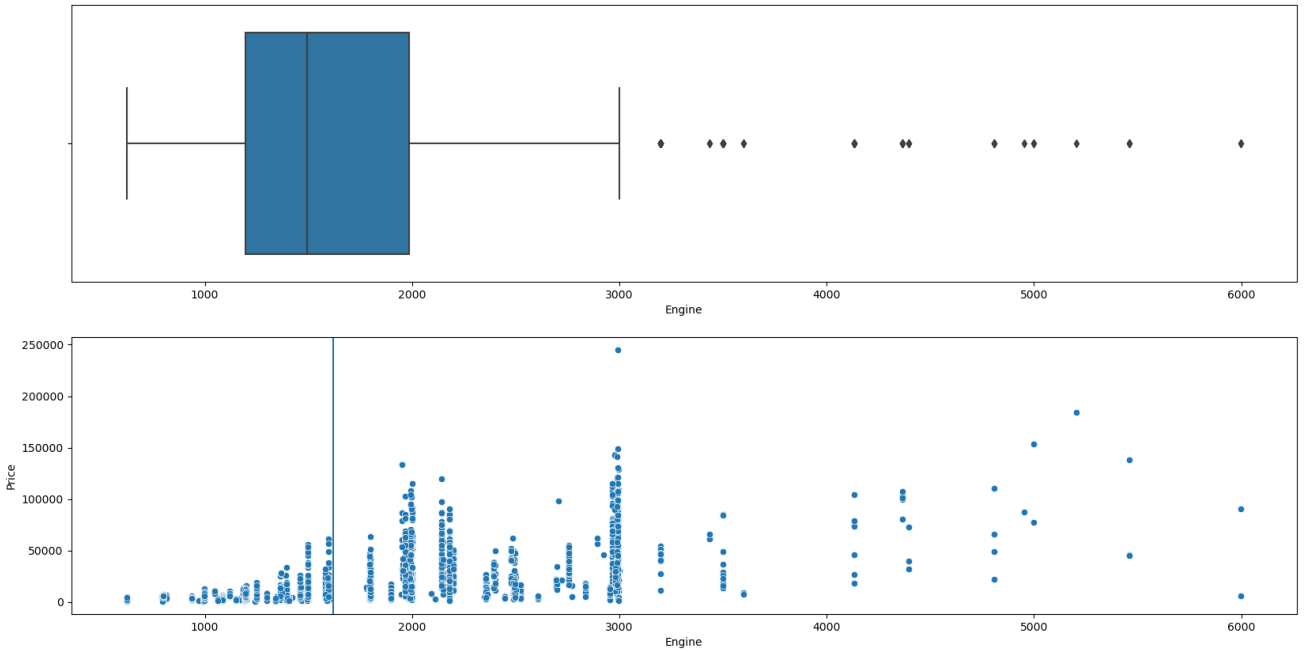


Price와 비교해본 결과 Mileage이 적을 수록 Price 가 높게 분포되는 경향이 있다.

따라서 Mileage 변수는 Price에 영향을 미친다고 할 수 있다.

5. Engine – 데이터 분포확인

	Engine
count	6153.000000
mean	1620.154884
std	601.334518
min	624.000000
25%	1198.000000
50%	1493.000000
75%	1984.000000
max	5998.000000



데이터들을 직접 확인하여 높은 배기량을 가지는 데이터는 도메인 지식을 활용한 결과 결측치나 이상치가 아닌 것으로 판단되었다.

따라서 현재 데이터의 과적합을 방지하고 데이터의 다양성을 유지하기 위해 EDA를 통해서는 이상치를 제거하지 않도록 하겠다.

5. Engine – 결측치 확인

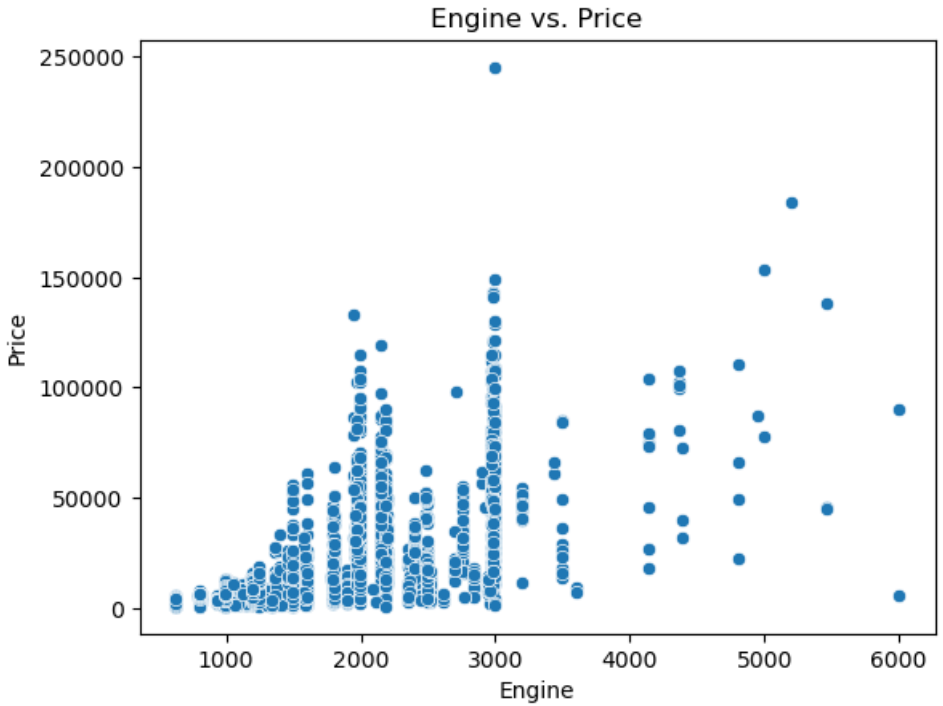
	Name	Location	Price	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price
	Honda City 1.5 GXI	Ahmedabad	8.416761	2007	60006	Petrol	Manual	First	12.800000	NaN	NaN	NaN	NaN
	Maruti Swift 1.3 VXi	Kolkata	8.081645	2010	42001	Petrol	Manual	First	16.100000	NaN	NaN	NaN	NaN
	Maruti Swift 1.3 VXi	Chennai	7.894572	2006	97800	Petrol	Manual	Third	16.100000	NaN	NaN	NaN	NaN
	Land Rover Range Rover 3.0 D	Mumbai	10.612101	2008	55001	Diesel	Automatic	Second	18.659539	NaN	NaN	NaN	NaN
	Honda City 1.3 DX	Delhi	8.498106	2009	55005	Petrol	Manual	First	12.800000	NaN	NaN	NaN	NaN

Engine 에 대한 결측치 처리는 또한 데이터의 손실을 최대한 줄여주기 위해서 자동차 회사와 이름이 같은 데이터가 존재하면 그 데이터의 값을 이용하여 대치시켜 주도록 하겠다.

	삭제 이전	삭제 이후
계	6189	6158

그 결과 36개의 데이터 중, 5개의 데이터를 대치하고 31개의 데이터를 삭제해 주었다.

5. Engine – Price 와 비교

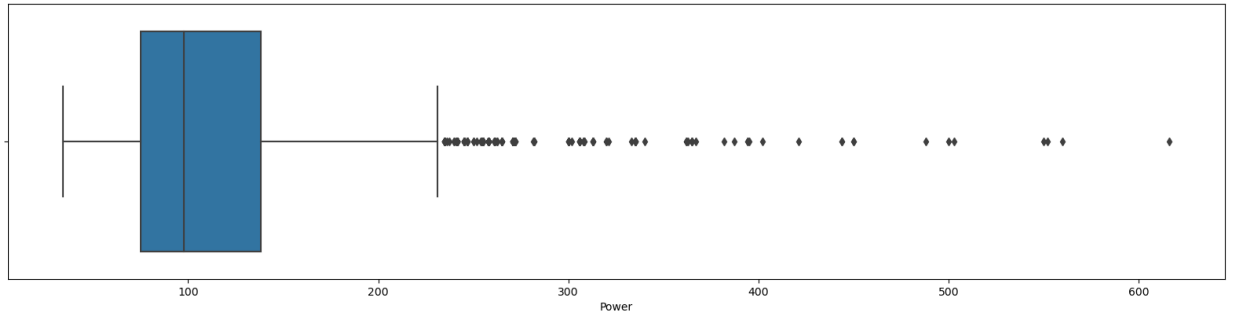


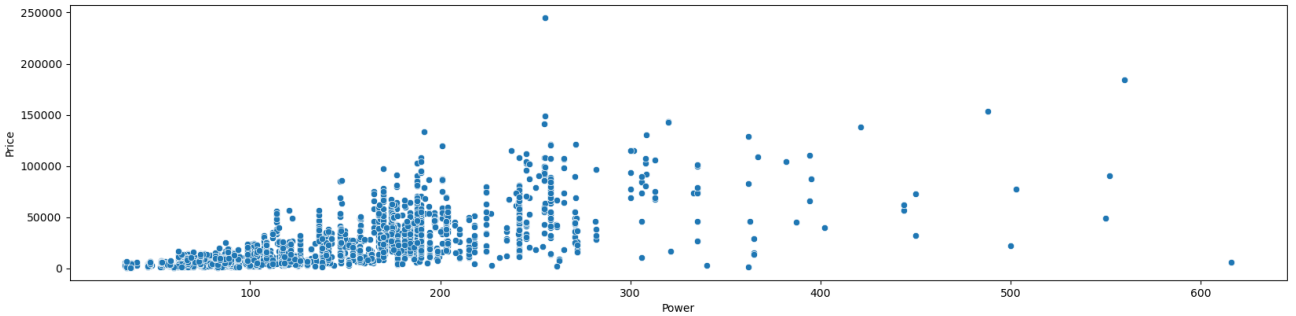
Price와 비교해본 결과 Engine이 증가할수록 Price 가 높게 분포되는 경향이 있다.

따라서 Engine 변수는 Price에 영향을 미친다고 할 수 있다.

6. Power – 데이터 분포 확인

	Power
count	6046.000000
mean	113.323118
std	54.166353
min	34.200000
25%	75.000000
50%	97.650000
75%	138.100000
max	616.000000





데이터들을 직접 확인하여 높은 마력을 가지는 데이터는 도메인 지식을 활용한 결과 결측치나 이상치가 아닌 것으로 판단되었다.

따라서 현재 데이터의 과적합을 방지하고 데이터의 다양성을 유지하기 위해 EDA를 통해서는 이상치를 제거하지 않도록 하겠다.

6. Power – 결측치 확인

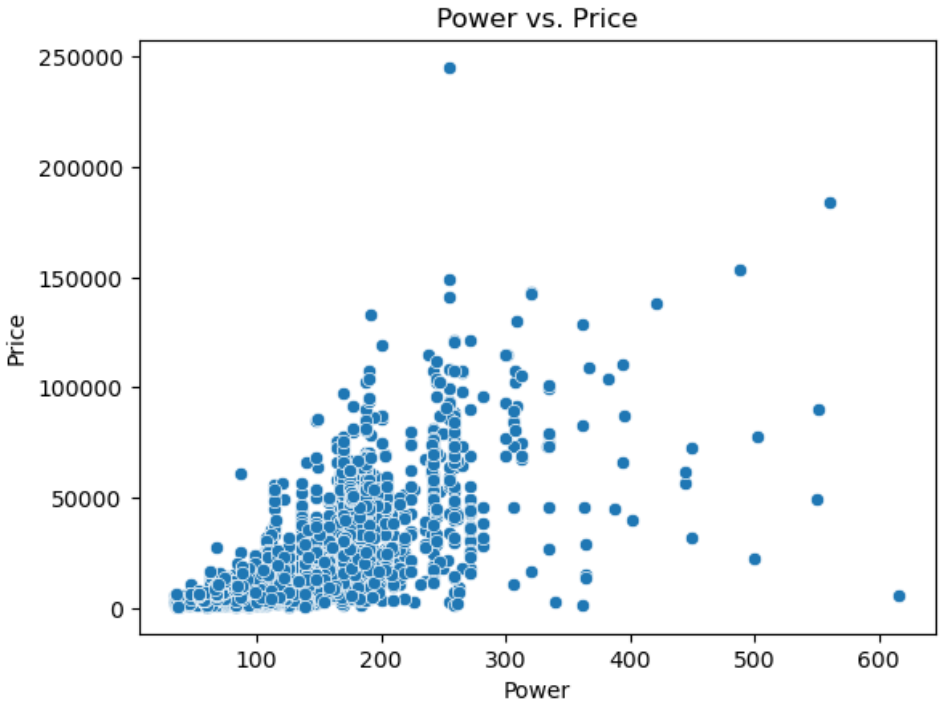
Name	Location	Price	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price
Ford Fiesta 1.4 SXi TDCi	Jaipur	8.028103	2008	111111	Diesel	Manual	First	17.800000	1399.0	NaN	5.0	NaN
Hyundai Santro Xing XL	Hyderabad	7.597321	2005	87591	Petrol	Manual	First	17.475237	1086.0	NaN	5.0	NaN
Hyundai Santro Xing XO	Hyderabad	8.076894	2007	73745	Petrol	Manual	First	17.000000	1086.0	NaN	5.0	NaN
Hyundai Santro Xing XL eRLX Euro III	Mumbai	7.172440	2005	102000	Petrol	Manual	Second	17.000000	1086.0	NaN	5.0	NaN
Hyundai Santro Xing XO eRLX Euro II	Kochi	7.847778	2008	80759	Petrol	Manual	Third	17.000000	1086.0	NaN	5.0	NaN

Power 에 대한 결측치 처리는 또한 데이터의 손실을 최대한 줄여주기 위해서 자동차 회사와 이름이 같은 데이터가 존재하면 그 데이터의 값을 이용하여 대치시켜 주도록 하겠다.

	삭제 이전	삭제 이후
계	6158	6154

그 결과 112개의 데이터 중, 108개의 데이터를 대치하고 4개의 데이터를 삭제해 주었다.

6. Power – Price 와 비교

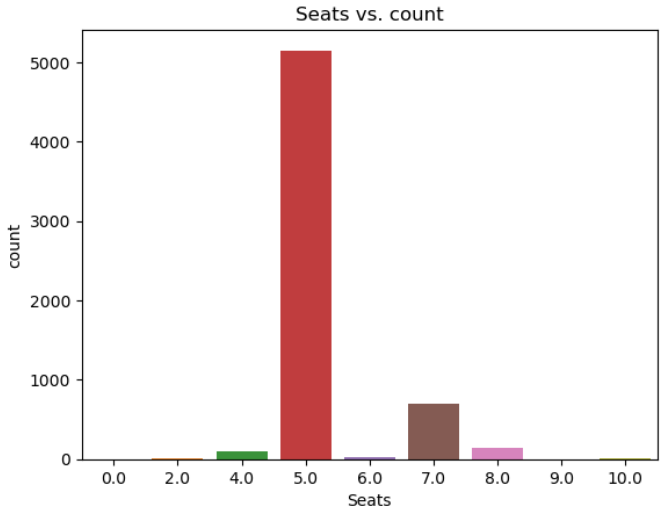


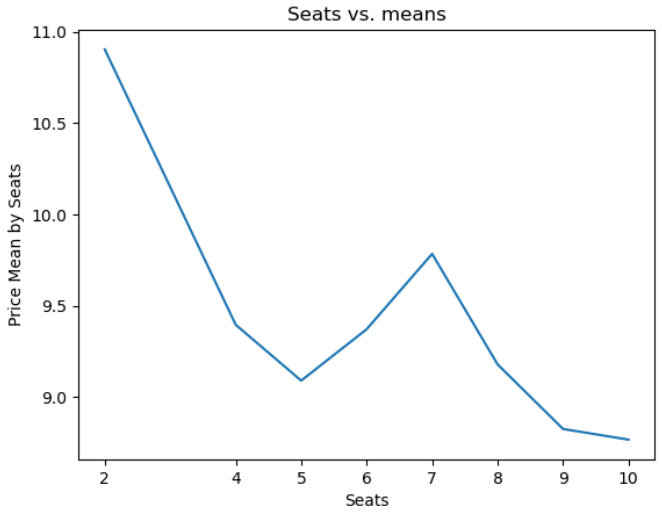
Price와 비교해본 결과 Power가 증가할수록 Price 가 높게 분포되는 경향이 있다.

따라서 Power 변수는 Price에 영향을 미친다고 할 수 있다.

7. Seats - 데이터 분포 확인

	Seats
count	6143.00000
mean	5.27918
std	0.80924
min	0.00000
25%	5.00000
50%	5.00000
75%	5.00000
max	10.00000





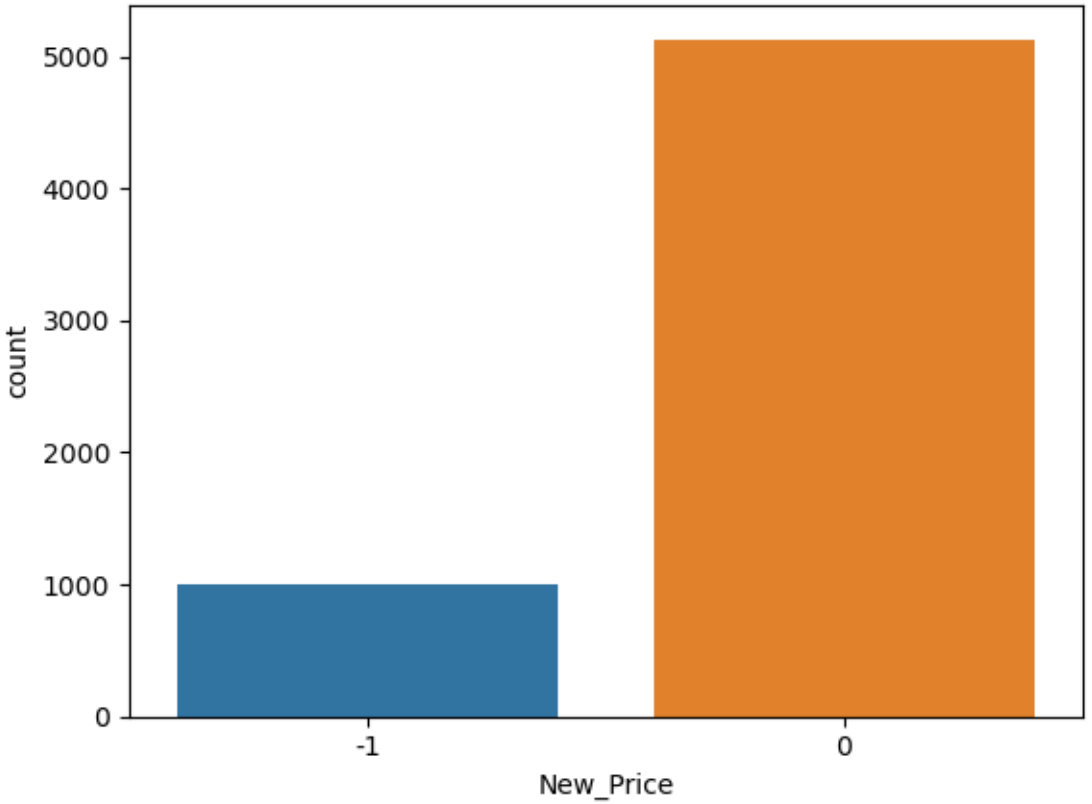
데이터 분포를 확인해본 결과 Seats 가 5개인 경우의 중고차가 제일 많았고 평균 가격은 그렇게 높지 않았다.

또한 Seats 수가 2인 경우 제일 데이터의 분포가 적었지만 평균 가격은 제일 높은 것으로 확인되었다.

따라서 Seats 의 수가 가격을 예측하는데 영향을 미친다고 볼 수 있다. 따라서 변수를 삭제하지 않겠다.

추후에 분석을 진행할 경우 이를 순서가 정해진 순서형 변수로 생각하여 분석을 진행하는 것이 좋을 것 같다

7. New_Price - 데이터 분포 확인



New_Price 데이터에 결측치가 5150가 존재하였다. 열을 바로 지워버릴 수 있었으나, 데이터 손실을 최소화 하고자
신차가 출고되면 중고차 가격이 내려갈 것이라는 도메인 지식을 활용하여 신차가 출고 되었을 경우 -1, 출고되지 않았을 경우 0으로 치환해 주었다.

7. Seats – 이상치 및 결측치 제거

이상치 제거

	Name	Seats	New_Price
3999	Audi A4 3.2 FSI Tiptronic Quattro	0.0	NaN

이상치가 1개 존재하여 모델에 큰 영향을 미치지 않을 것으로 생각되어 삭제하였다.

결측치 제거

	Name	Location	Price	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price
194	Honda City 1.5 GXI	Ahmedabad	8.416761	2007	60006	Petrol	Manual	First	12.800000	1493.0	98.6	NaN	NaN
229	Ford Figo Diesel	Bangalore	8.615890	2015	70436	Diesel	Manual	First	18.659539	1498.0	99.0	NaN	NaN
1385	Honda City 1.5 GXI	Pune	7.740421	2004	115000	Petrol	Manual	Second	12.800000	1493.0	184.0	NaN	NaN
1917	Honda City 1.5 EXI	Jaipur	7.865583	2005	88000	Petrol	Manual	Second	13.000000	1493.0	100.0	NaN	NaN
2264	Toyota Etios Liva V	Pune	8.416761	2012	24500	Petrol	Manual	Second	18.300000	1197.0	174.5	NaN	NaN

데이터의 결측치를 이전 방식과 같이 데이터 내에 이름이 존재할 경우 그 데이터의 값으로 대치시켜주고, 이름이 존재하지 않았을 경우 삭제 하도록 하겠다.

그 결과 11개의 데이터 중 11개를 대치 시키고 0개의 데이터를 삭제 시켰다.

1. Name

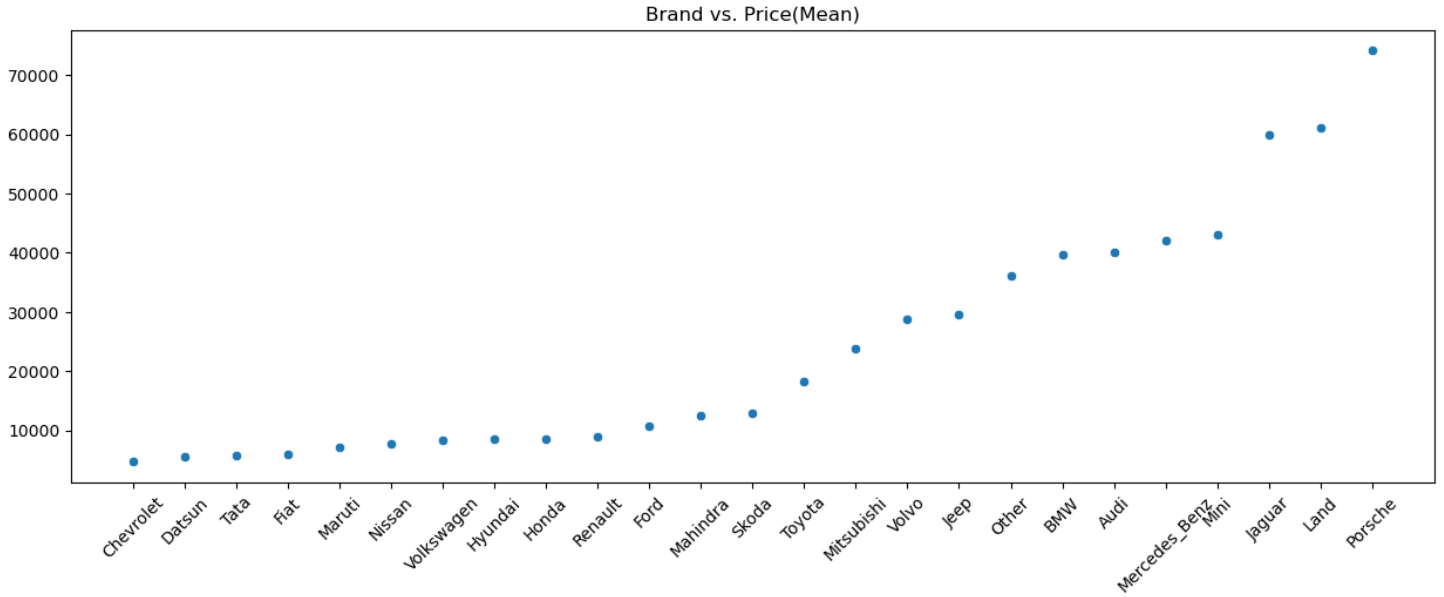
	Brand
0	Maruti
1	Hyundai
2	Honda
3	Maruti
4	Audi
5	Hyundai
6	Nissan
7	Toyota
8	Volkswagen
9	Tata

자동차 이름과 회사가 같이 있는 데이터는 활용할 수 없다고 판단하여 자동차를 브랜드 별로 구분하기 위해 브랜드만 가져왔다.

	count
Volvo	21
Jeep	19
Porsche	18
Datsun	17
ISUZU	3
Force	3
Bentley	2
Smart	1
Ambassador	1
Lamborghini	1

또한 자동차 브랜드의 개수가 적은 데이터가 존재하였고, Datsun 보다 개수가 적은 자동차들은 우리가 많이 보유하고 있지 않은 자동차라고 생각하여, 주된 고객층을 확보하기 위해서 Other로 한꺼번에 분류하도록 하겠다.

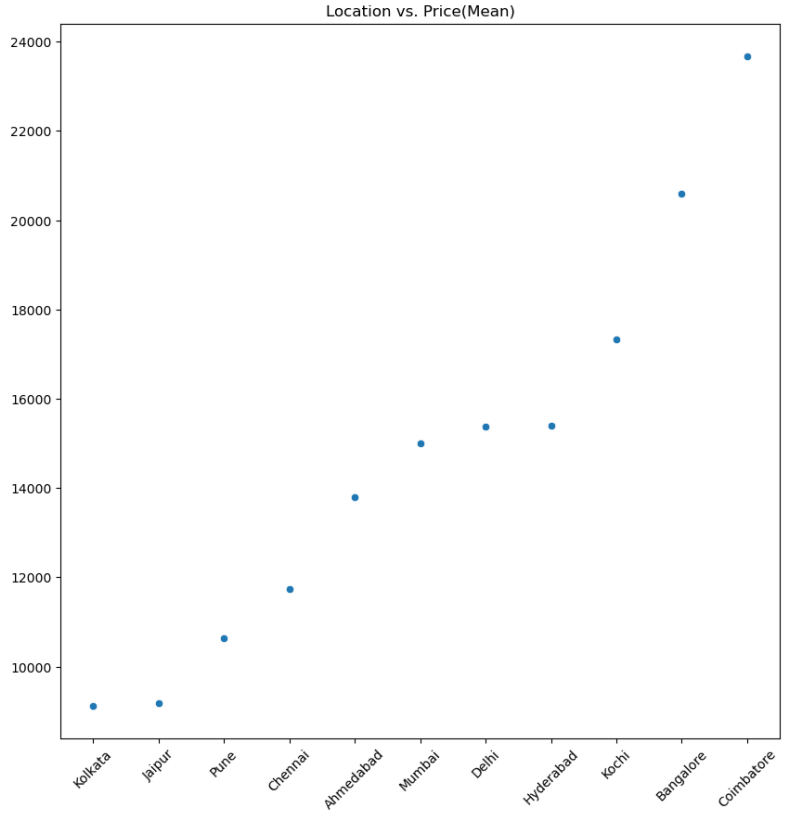
1. Name – Price와 비교

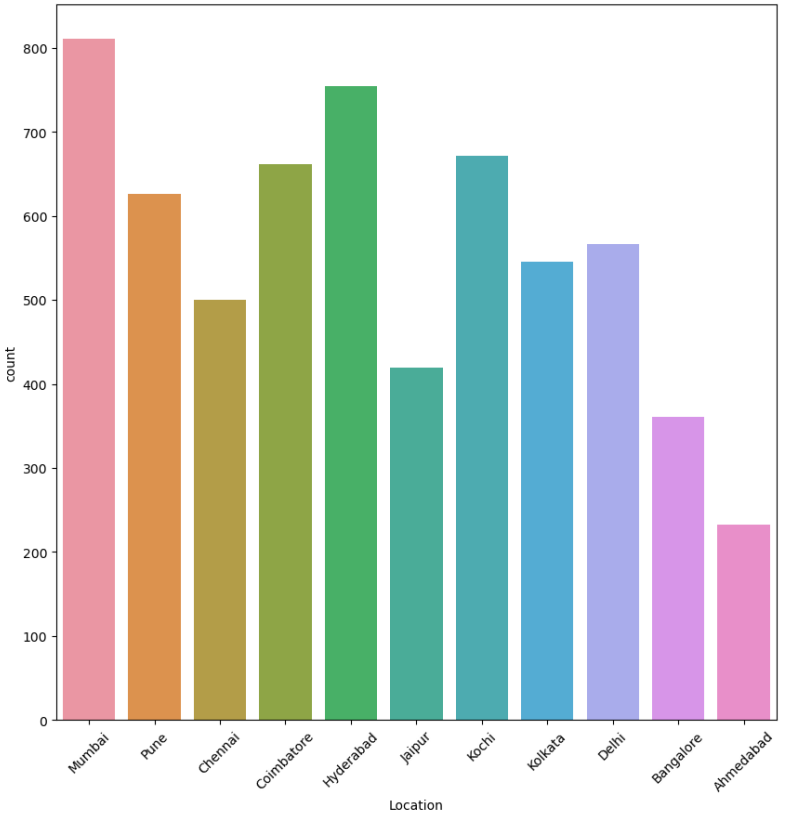


자동차 브랜드별 판매 가격의 평균을 확인해본 결과 자동차 브랜드별로 가격차이가 있는 것으로 확인되었다.

또한 도메인 지식을 활용하여 결과를 확인한 결과 브랜드 가치가 높은 차량의 가격이 상대적으로 높게 분포된 것을 확인할 수 있다. 따라서 brand 이름이 Price에 영향을 미친다고 할 수 있다.

2. Location – Price와 비교

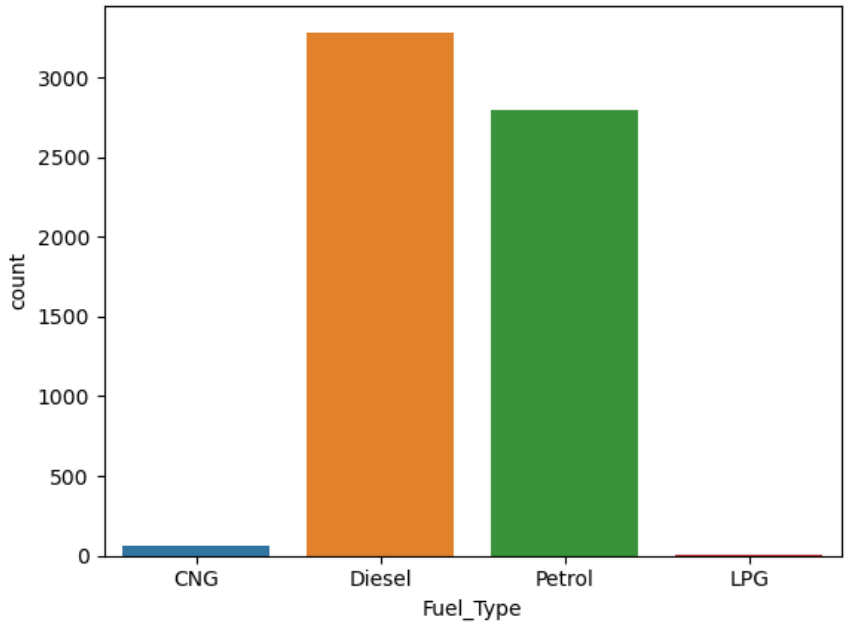




Location 별로 Price의 평균과 자동차의 보유 현황이 다른 것을 확인할 수 있다.

따라서 지역과 자동차 가격의 연관성을 찾을 수만 있다면 지점을 개점할 때 판매 수익을 높일 수 있는 지점을 찾아 열 수 있을 것으로 보인다.

3. Fuel_Type



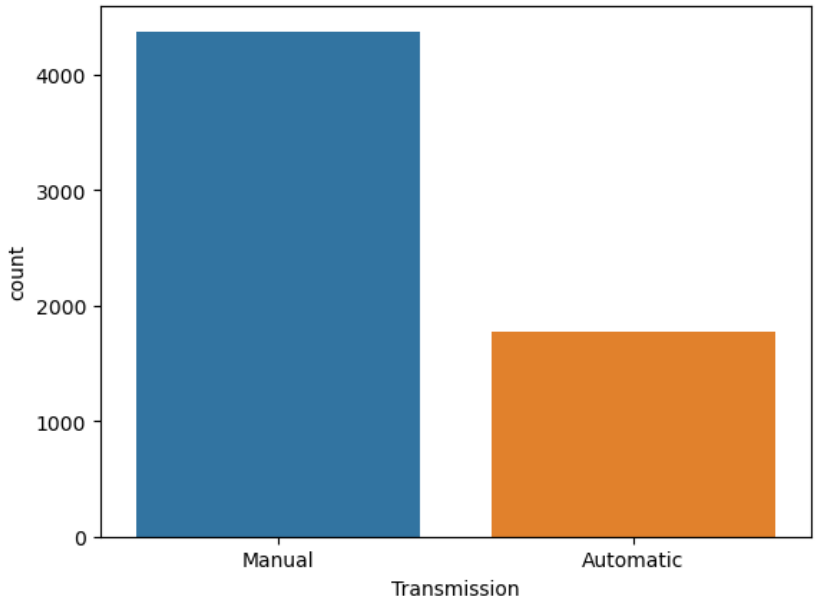
	Price	Year	Kilometers_Driven	Mileage	Engine
CNG	5421.837193	2014.052632	55052.263158	25.538070	1089.421053
Diesel	20131.506102	2013.825320	64461.042301	18.812894	1858.882532
Petrol	9054.679986	2013.022500	47177.164643	17.742938	1351.794286

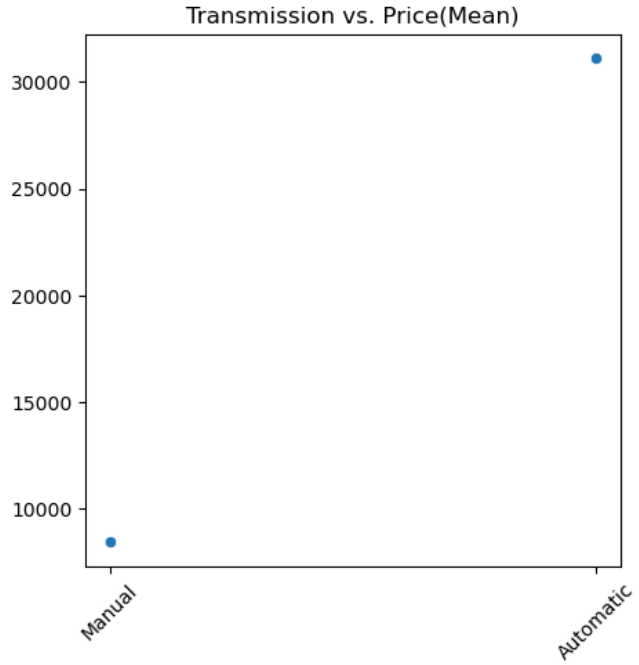
- CNG와 LPG 가 Diesel과 Petrol에 비해서 매우 적은 비율을 차지하고 있다.
- LPG는 데이터 개수가 30을 넘지 못하므로 모델에 전혀 영향을 미칠 수 없다고 생각되어 지우도록 하겠다.

- CNG의 경우 Price가 높은 편은 아니나 Mileage 가 가장 높게 나와 경제적인 자동차의 특징을 보이고 있다.
- Diesel의 경우 가격이 가장 높은 편이고, Kilometers_Driven 또한 가장 높게 나왔다.
- Petrol은 Kilometers_Driven이나 Mileage가 가장 낮게 나타나고 있다.

따라서 Fuel_Type 변수는 Price에 영향을 미치고, 다중공선성이 존재할 가능성이 있다.

4. Transmission

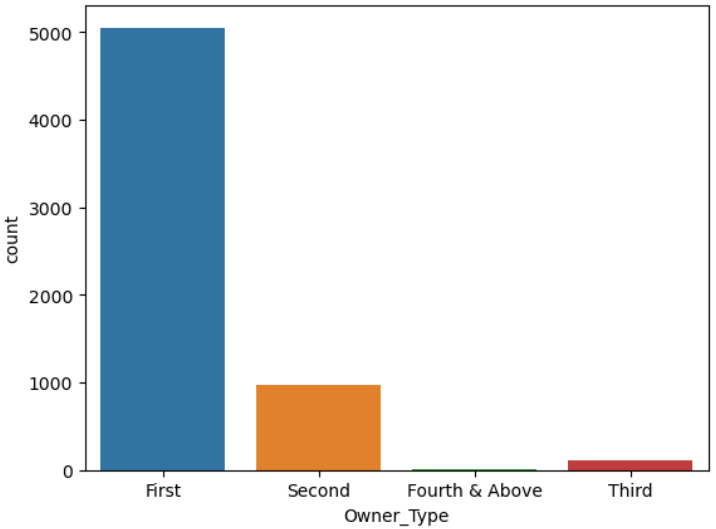


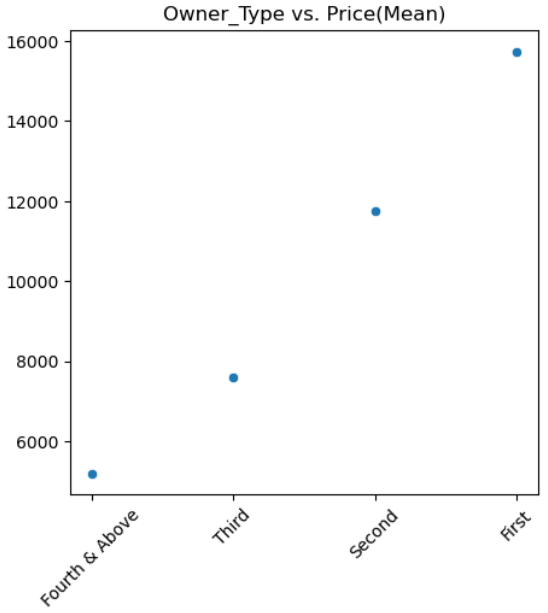


Transmisson 의 종류에 따라 가격의 변화가 있다는 것을 확인할 수 있다.

- Manual의 경우 개수가 많으나 가격의 평균은 낮은 것으로 보인다.
- Automatic의 경우 개수가 적으나 가격의 평균이 높은 것으로 보인다.
- 따라서 Automatic의 경우가 비싼 경향이 있어 보인다.

5. Owner_Type

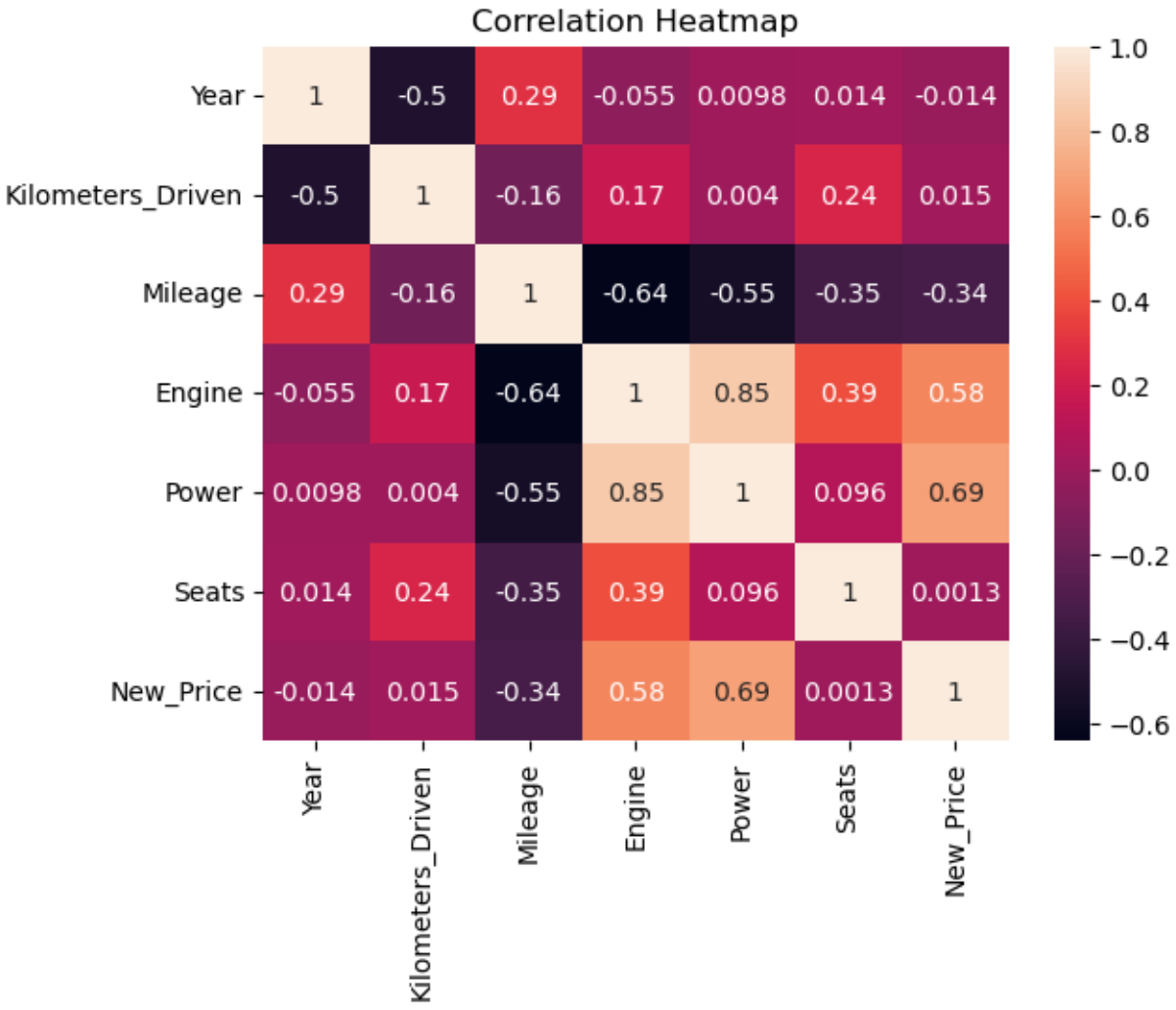




Owner Type 에 따라 평균 가격이 달라진다고 생각된다. 따라서 Price에 영향을 주는 변수라고 생각된다.

하지만 Fourth & Above의 숫자가 너무 적고 평균 가격도 이상치 처럼 낮기 때문에 모델에 악영향을 주거나 아예 영향을 끼치지 못할 것이라고 판단되어 삭제해 주도록 하겠다.

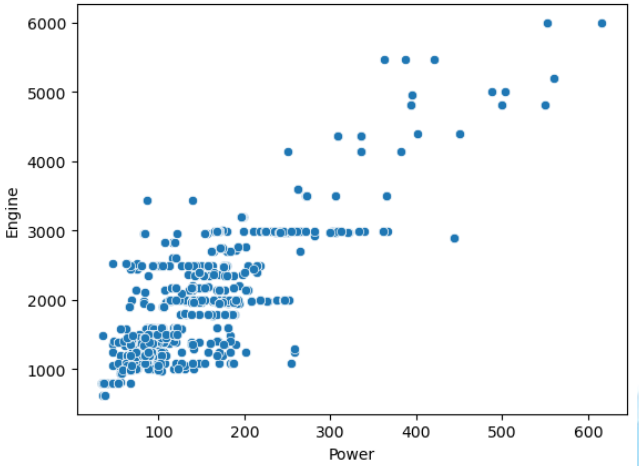
숫자형 데이터 상관관계 파악



판단 기준은 correlation 이 0.7 이상인 변수만 상관성이 높다고 생각한다.

따라서 Power와 Engine 둘의 상관계수가 0.85로 매우 높은 양의 상관관계에 있고, 그래프를 확인한 결과 선형 상관관계에 있는 것을 확인하였다.

따라서 Engine과 Power 둘 중 다른 변수들과의 상관성이 높은 Engine 변수를 제거하여 다중공선성을 없애도록 하겠다.



범주형 데이터 상관관계 파악

```python
# 크로스 테이블 생성
_temp = df.columns.to_list()
columns = df.columns
result = []
for i in columns:
    for j in _temp:
        if i != j:
            cross_table = pd.crosstab(df[i], df[j])
            # 카이제곱 검정 실행
            chi2, p, dof, expected = chi2_contingency(cross_table)
            # 결과 출력
            if p > 0.05:
                print("{} 와 {} 간의 카이제곱 검정 결과 : p-value = {:.5f}".format(i,j,p))
                result.append(i)
                result.append(j)
    _temp.remove(i)
```

✓ 1.8s

```
Brand 와 Kilometers_Driven 간의 카이제곱 검정 결과 : p-value = 1.00000
Location 와 Kilometers_Driven 간의 카이제곱 검정 결과 : p-value = 0.66643
Price 와 Owner_Type 간의 카이제곱 검정 결과 : p-value = 1.00000
Year 와 Kilometers_Driven 간의 카이제곱 검정 결과 : p-value = 1.00000
Kilometers_Driven 와 Transmission 간의 카이제곱 검정 결과 : p-value = 0.79797
Kilometers_Driven 와 Owner_Type 간의 카이제곱 검정 결과 : p-value = 1.00000
Kilometers_Driven 와 Mileage 간의 카이제곱 검정 결과 : p-value = 1.00000
Kilometers_Driven 와 Power 간의 카이제곱 검정 결과 : p-value = 1.00000
Fuel_Type 와 New_Price 간의 카이제곱 검정 결과 : p-value = 0.94999
Transmission 와 Owner_Type 간의 카이제곱 검정 결과 : p-value = 0.22664
```

범주형 자료들 간의 카이제곱 검정을 한 결과 카이제곱 검정의 p-value가 0.05보다 높지 못한 변수는 Seats 인 것으로 나타났다 따라서 검정 결과 Seats 변수를 삭제해주도록 하겠다.

데이터 스케일링

	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Power	New_Price
0	-1.00	0.507493	CNG	Manual	First	1.450877	-0.626624	0.0
1	0.25	-0.287348	Diesel	Manual	First	0.235088	0.451664	0.0
2	-0.75	-0.159148	Petrol	Manual	First	-0.022807	-0.142631	-1.0
3	-0.50	0.892094	Diesel	Manual	First	0.428070	-0.141680	0.0
4	-0.25	-0.295809	Diesel	Automatic	Second	-0.549123	0.683043	0.0
6	-0.25	0.892068	Diesel	Manual	First	0.833333	-0.548336	0.0
7	0.50	-0.415548	Diesel	Automatic	First	-1.222807	1.169572	-1.0
8	-0.25	0.313398	Diesel	Manual	First	0.387719	0.093502	0.0
9	-0.50	0.351910	Diesel	Manual	Second	0.696491	-0.375594	0.0
10	1.00	-0.679846	Petrol	Manual	First	0.566667	0.087956	-1.0

데이터의 혹시 모를 이상치에 민감하게 반응하지 않도록 하기 위해 RobustScaler를 실시해주도록 하겠다.

```
1  df_train_x, df_test_x , df_train_y, df_test_y = train_test_split(df_x,df_y,test_size = 0.3, random_state = 42)
2  print('y : {} , x : {}'.format(df_train_y.shape , df_train_x.shape))
3  print('y : {} , x : {}'.format(df_test_y.shape , df_test_y.shape))
```

```
y : (4294,) , x : (4294, 8)
y : (1841,) , x : (1841,)
```

Test 데이터와 train 데이터는 2:8의 비율로 나눠주도록 하겠다.

	Variables	VIF
0	const	1.430859
1	Year	1.581234
2	Kilometers_Driven	1.362201
3	Mileage	1.642596
4	Power	1.505623
5	New_Price	1.137968

연속형 변수들 간의 다중공선성은 없는 것으로 보인다.
(scaling 의 영향일 가능성이 있다.)

	Coefficient	t-statistics	p-values
Intercept	9.71	180.08	0.00
Brand[T.BMW]	-0.03	-1.14	0.26
Brand[T.Chevrolet]	-0.94	-23.62	0.00
Brand[T.Datsun]	-0.88	-10.03	0.00
Brand[T.Fiat]	-0.90	-11.93	0.00
Brand[T.Ford]	-0.69	-21.38	0.00
Brand[T.Honda]	-0.52	-17.94	0.00
Brand[T.Hyundai]	-0.61	-21.54	0.00
Brand[T.Jaguar]	0.11	2.05	0.04
Brand[T.Jeep]	-0.44	-5.41	0.00
Brand[T.Land]	0.20	4.20	0.00
Brand[T.Mahindra]	-0.66	-20.52	0.00
Brand[T.Maruti]	-0.60	-20.34	0.00
Brand[T.Mercedes_Benz]	0.04	1.42	0.15
Brand[T.Mini]	0.42	6.75	0.00
Brand[T.Mitsubishi]	-0.26	-4.35	0.00
Brand[T.Nissan]	-0.63	-15.13	0.00
Brand[T.Other]	-0.82	-7.78	0.00
Brand[T.Porsche]	0.36	4.38	0.00
Brand[T.Renault]	-0.67	-17.41	0.00
Brand[T.Skoda]	-0.50	-15.03	0.00
Brand[T.Tata]	-1.09	-30.32	0.00
Brand[T.Toyota]	-0.32	-10.84	0.00
Brand[T.Volkswagen]	-0.65	-20.42	0.00
Brand[T.Volvo]	-0.23	-3.17	0.00
Location[T.Bangalore]	0.11	4.12	0.00
Location[T.Chennai]	0.00	0.12	0.91
Location[T.Coimbatore]	0.06	2.28	0.02
Location[T.Delhi]	-0.09	-3.43	0.00
Location[T.Hyderabad]	0.09	3.73	0.00
Location[T.Jaipur]	-0.09	-3.27	0.00
Location[T.Kochi]	-0.05	-2.16	0.03
Location[T.Kolkata]	-0.24	-9.39	0.00
Location[T.Mumbai]	-0.06	-2.60	0.01
Location[T.Pune]	-0.05	-2.03	0.04
Fuel_Type[T.Diesel]	0.27	6.24	0.00
Fuel_Type[T.Petrol]	-0.12	-2.71	0.01
Transmission[T.Manual]	-0.14	-9.89	0.00
Owner_Type[T.Second]	-0.06	-5.12	0.00
Owner_Type[T.Third]	-0.14	-4.11	0.00
Year	0.50	67.11	0.00
Kilometers_Driven	-0.05	-7.07	0.00
Mileage	-0.19	-19.66	0.00
Power	0.32	34.00	0.00
New_Price	-0.09	-7.25	0.00

Test 결과 범주형 데이터들에게서 Price에 영향을 끼치지 못하는 변수들이 여럿 존재하였다.

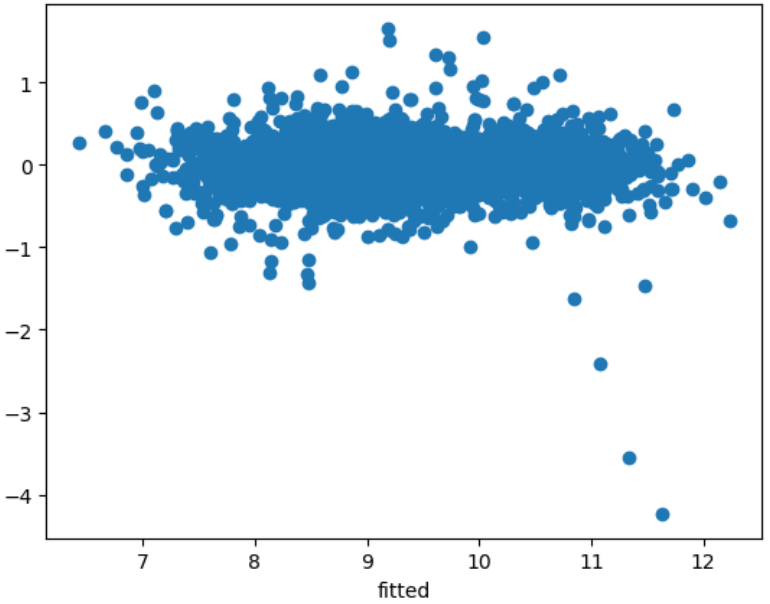
Brand의 경우 BMW, Mercedes_Benz

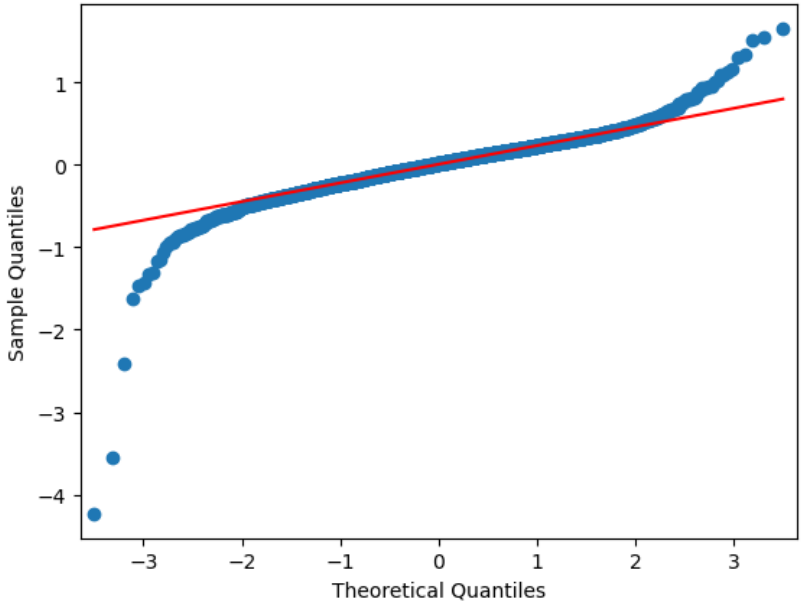
Location의 경우 Chennai

pvalue 가 높게 나와 영향을 끼치지 못했다.

따라서 만일 사업을 할 경우 BMW와 Mercedes Benz는 피하고,

Chennai 지역은 피해서 창업을 할 경우 이 모델이 어느정도 예측을 할 수 있을 것으로 보인다.

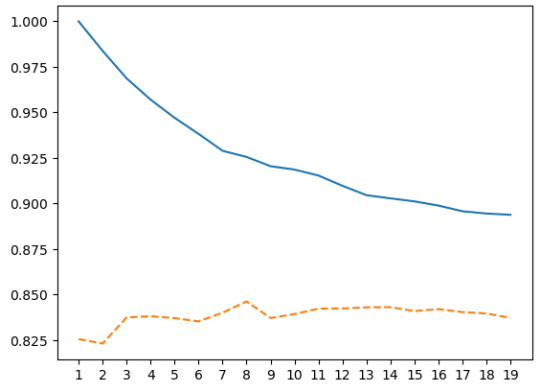




잔차 그래프와 qqplot을 확인해본 결과 잔차의 독립성과 정규성을 만족하는 것 처럼 보이진 않는다.

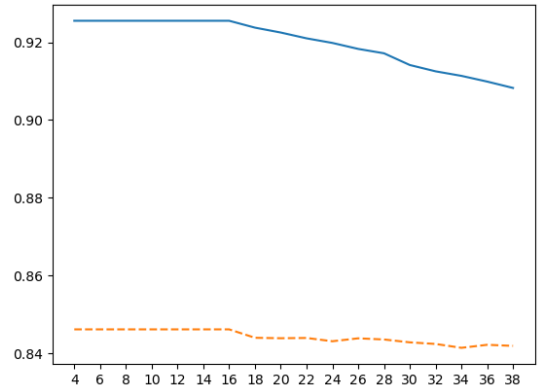
따라서 추가적인 데이터 전처리 혹은 데이터 수집을 통하여 부족한 데이터를 채워주는 것이 좋을 것 같다.

min_samples_leaf



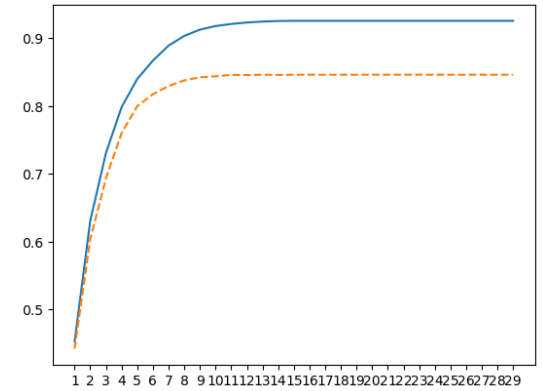
Train score가 계속해서 감소하고 있고,
Test score 가 8에서 증가한뒤, 감소하였으므로 8을 선택하도록 하겠다.

min_samples_split



Train score 와 test score가 16까지 일정하다가 그 이후에 떨어지고 있다. 따라서 16이전인 14을 선택하도록 하겠다

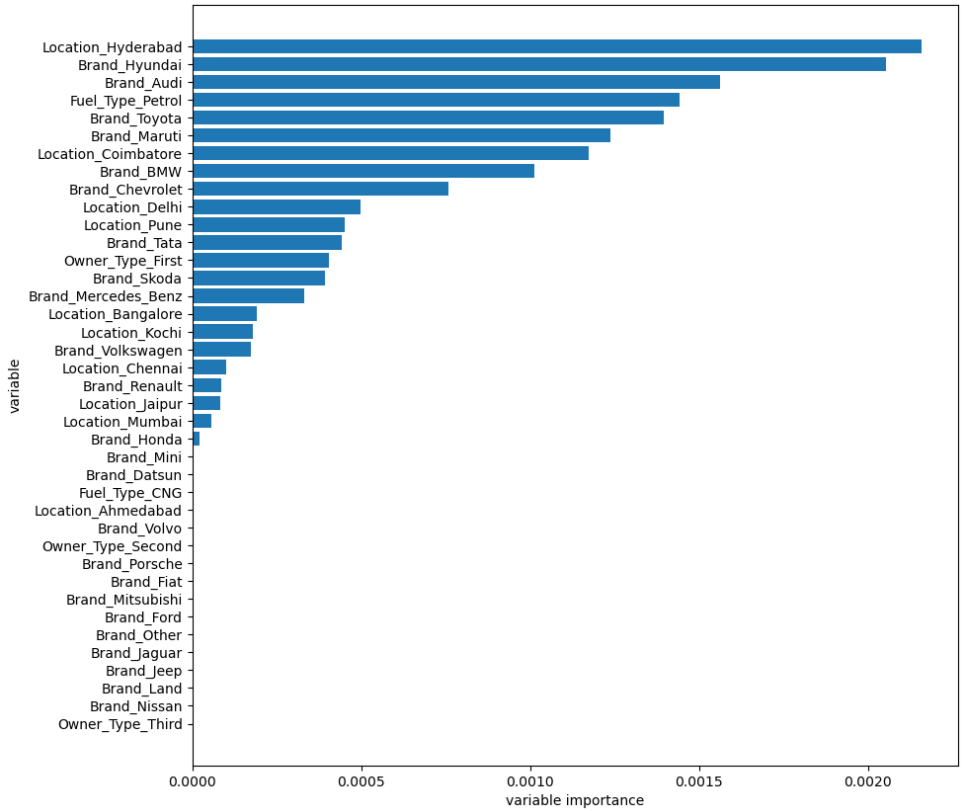
max_depth



Train score와 Test score 가 동일한 양상을 보이고 있다.

Test score가 13에서 가장 높았으므로 13을 선택하도록 하겠다.

	Parameters
ccp_alpha	0.0
criterion	squared_error
max_depth	13
max_features	None
max_leaf_nodes	None
min_impurity_decrease	0.0
min_samples_leaf	8
min_samples_split	16
min_weight_fraction_leaf	0.0
random_state	42
splitter	best

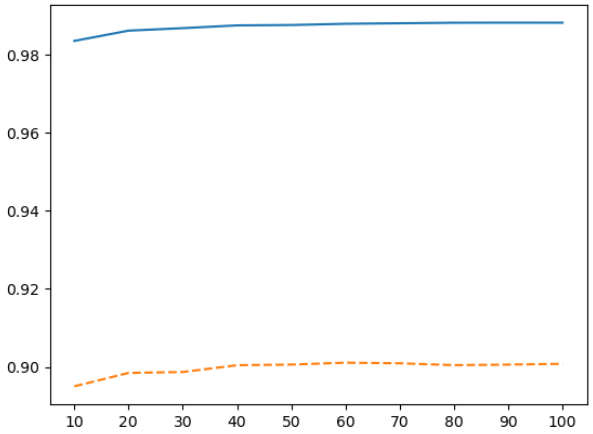


최종 모형은 위와같이 나왔고, 변수의 중요도는 Location, Brand, Fuel_Type 등이 제일 중요한 변수들이라고 나왔다.

	Train	Test
Linear_Regression	0.895294	0.893514
Decision_Tree	0.92447	0.846116

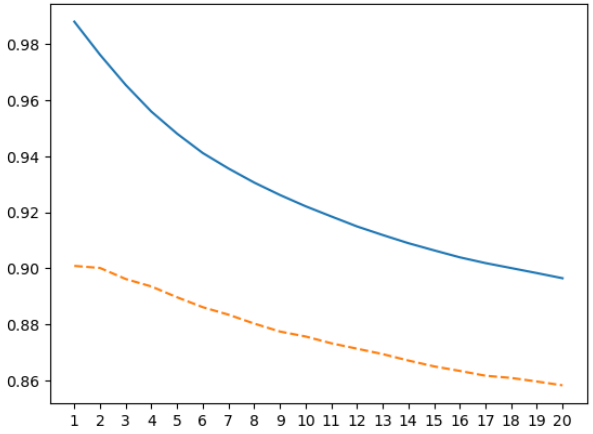
Test score는 0.83으로 회귀분석 보다 더 낮게 나왔다.
이 모델의 정확도는 매우 낮게 나온 편이니 선택하지 않도록 하겠다.
또한 linear regression 에서 확인한 변수들 중 Year 가 크게 나왔으나 이곳에서는 보이지 않았다.
따라서 Decision Tree가 정확한 예측을 해줬다고 할 수 없다.

N_estimator



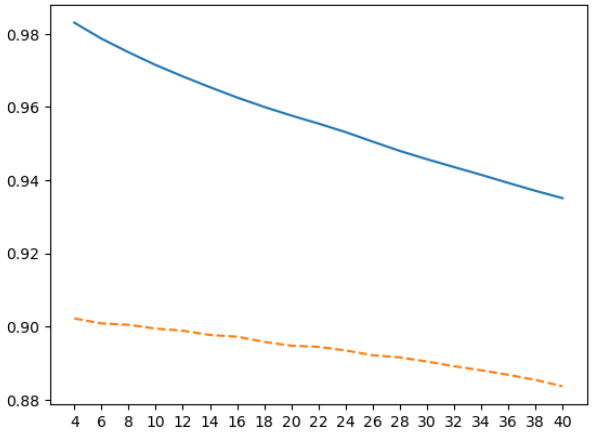
Train score 와 Test score 가 동일한 모양을 나타내고 있고 가장 gap 이 좁은 70을 선택하도록 하겠다.

min_samples_leaf



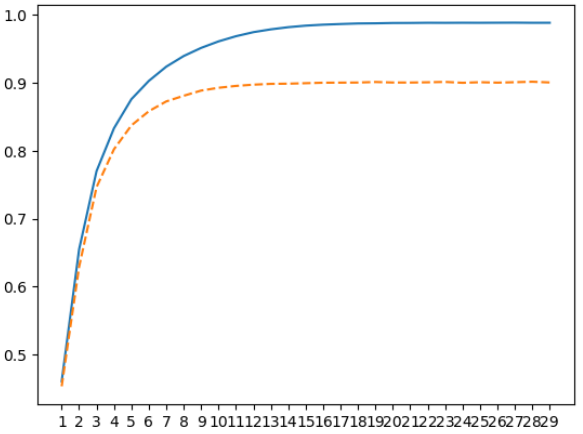
Train score 와 Test score 가 동일한 모양으로 점점 줄어드므로, 지정해주지 않도록 하겠다.

min_samples_split



Train score 와 Test score 가 동일한 모양으로 점점 줄어드므로, 지정해주지 않도록 하겠다.

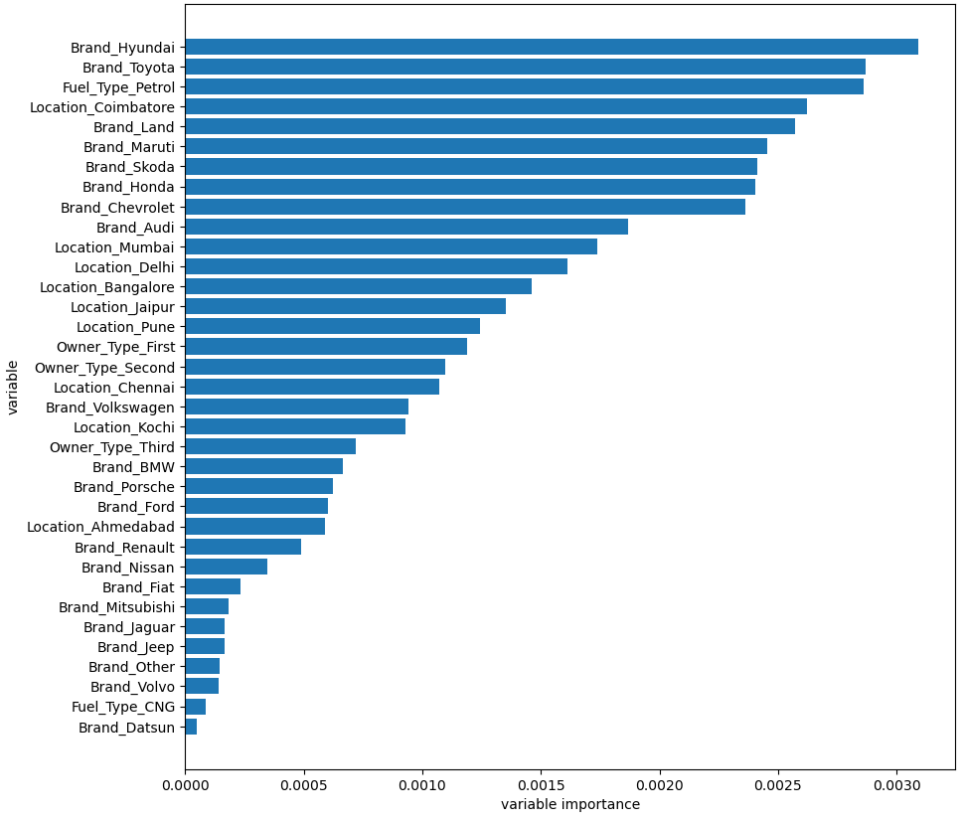
max_depth



Train score 와 Test score 가 동일한 모양을 나타내고 있고 가장 Test score 가 높은 28을 선택하도록하겠다.

3. Random Forest

	Parameters
bootstrap	True
ccp_alpha	0.0
criterion	squared_error
max_depth	28
max_features	1.0
max_leaf_nodes	None
max_samples	None
min_impurity_decrease	0.0
min_samples_leaf	1
min_samples_split	2
min_weight_fraction_leaf	0.0
n_estimators	70
n_jobs	None
oob_score	False
random_state	42
verbose	0
warm_start	False



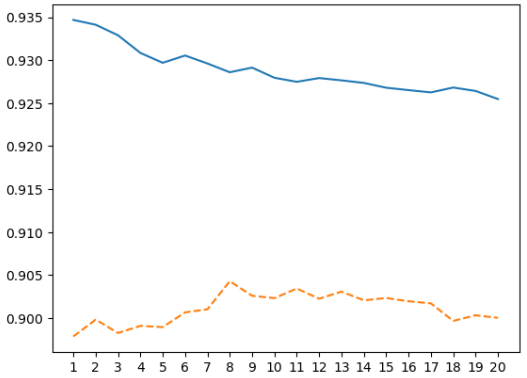
최종 모형은 위와같이 나왔고, 변수의 중요도는 Location, Brand, Fuel_Type 등이 제일 중요한 변수들이라고 나왔다.

	Train	Test
Linear_Regression	0.895294	0.893514
Decision_Tree	0.92447	0.846116
Random_Forest	0.987983	0.90133

Test score는 0.90으로 회귀변수 보다 높게 나왔다.
이 모델의 정확도는 낮게 나온 편이니 선택하지 않도록 하겠다.
또한 linear regression 에서 확인한 변수들 중 Year 카 크게 나왔으나 이곳에서는 보이지 않았다.
범주형으로 One-Hot encoding 을 해서 그런지 숫자형 변수들의 연관성이 제대로 보이지 않는 것 같다.

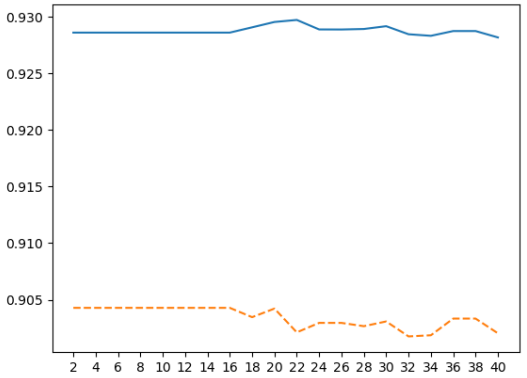
4. Gradient Boosting

min_samples_leaf



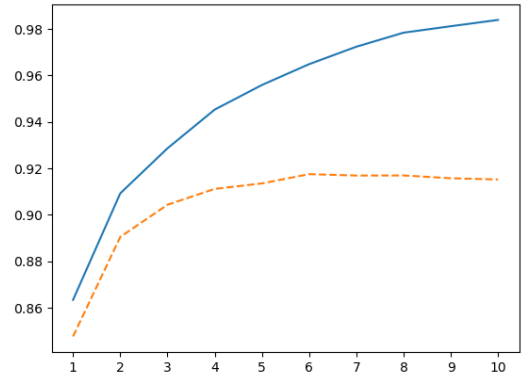
Train score가 계속해서 감소하고 있고,
Test score 가 8에서 증가한뒤, 감소하였으므로 8을 선택하도록 하겠다.

min_samples_split



Test score 의 크기가 변하지 않고 gap 이 가장 적은 12를 선택

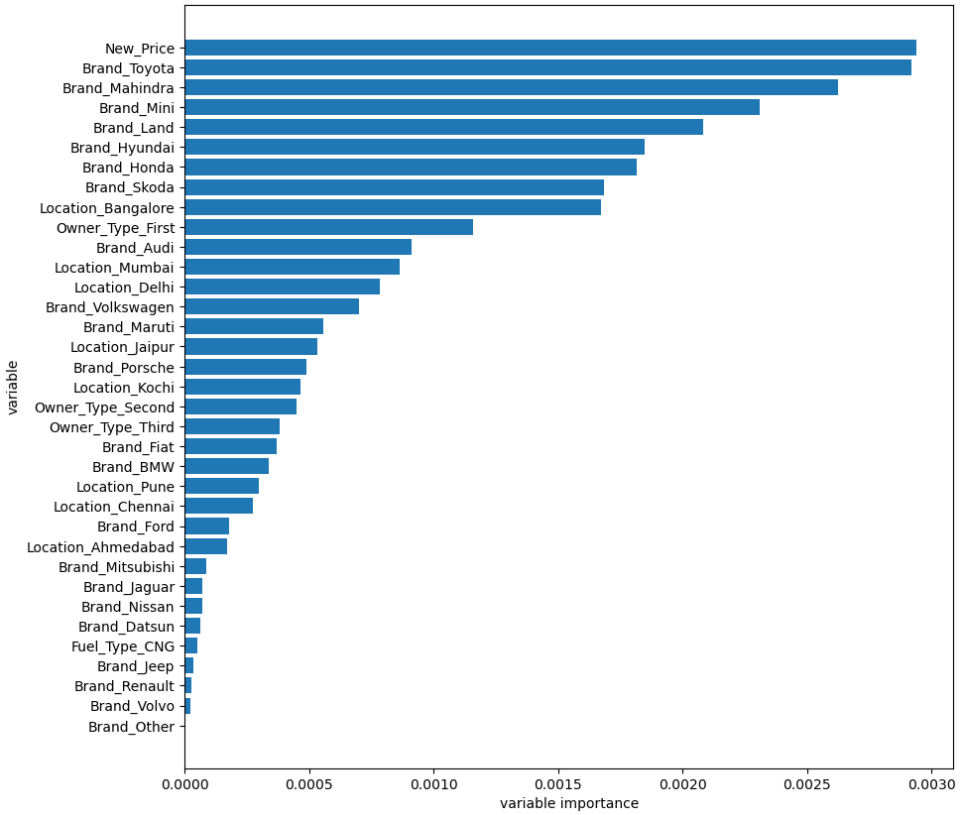
max_depth



Train score 와 Test score 가 일정하게 증가하고 있으므로 차이가 적고 Test score 가 높은 6을 선택해 주도록 하겠다.

4. Gradient Boosting

	Parameters
alpha	0.9
ccp_alpha	0.0
criterion	friedman_mse
init	None
learning_rate	0.25
loss	squared_error
max_depth	6
max_features	None
max_leaf_nodes	None
min_impurity_decrease	0.0
min_samples_leaf	8
min_samples_split	12
min_weight_fraction_leaf	0.0
n_estimators	120
n_iter_no_change	None
random_state	42
subsample	1.0
tol	0.0001
validation_fraction	0.1
verbose	0
warm_start	False



최종 모형은 위와같이 나왔고, 변수의 중요도는 New_Price, Brand, Owner_Type 등이 제일 중요한 변수들이라고 나왔다.

	Train	Test
Linear_Regression	0.895294	0.893514
Decision_Tree	0.92447	0.846116
Random_Forest	0.987983	0.90133
Gradient_Boosting	0.982385	0.922013

Test score가 4모델중 가장 높게 나왔다. 하지만 Train score 와의 차이가 너무 커서 과적합 된 것으로 확인된다.
New_Price와 Brand가 주요 인자로 선택 되었다.

결론 및 대안제시

	Train	Test
Linear_Regression	0.895294	0.893514
Decision_Tree	0.92447	0.846116
Random_Forest	0.987983	0.90133
Gradient_Boosting	0.982385	0.922013

Linear Regression , Decision Tree, Random Forest, Gradient Boosting 을 이용한 결과 Gradient_Boostin 모델의 정확도가 제일 높은 것으로 나타났다.

또한 각각의 변수 중요도를 확인해본 결과 종합적으로 Brand, Location, Fuel Type, Owner Type, New Price 가 제일 중요한 요소들로 작용되는 것을 확인할 수 있었다.

따라서 중고차 가격을 예측하기 위해서는 차량의 Brand와 이전 소유주의 Location 정보, 연료 타입, 중고차 구매 이력과 신차 출시 여부 등을 파악하여 미리 자동차 가격을 예측하여 소비자들에게 경제적이고 최대한 사업 이익을 얻을 수 있는 프로세스를 만들 수 있을 것이라 기대된다.

실습 과정을 통해 배운 또는 느낀 통찰, 아이디어, 애로사항 등을 정리합니다

통찰

1. 전처리 과정이 데이터 분석의 80%라는 것을 이론을 통해 배웠지만, 와닿지는 않았는데, 실제 분석을 하는 과정에서 데이터를 전처리하고, EDA하는 과정이 매우 어려운 것을 깨닫게 되었다.

   따라서 파이썬 기초 스킬들을 갈고 닦아 기초적인 데이터 정제를 빠르게 처리할 수 있는 능력을 길러야 겠다는 생각이 들었다.

2. 이번 과정에서 데이터 개수가 적어 데이터 손실을 최대한 줄이기 위해서 노력했는데, 같은 데이터 안에 있는 데이터들을 최대한 활용하고 도메인 지식들을 최대한 활용하는 것의 중요성을 깨닫게 되었다.

3. 데이터를 정제하는 과정에서 결측치의 형태가 한가지 형태만 있는 것이 아니라는 것을 깨닫게 되었다.
   따라서 단순히 함수를 입력하여 데이터를 요약하여 보지 말고, raw 데이터 그 자체를 들여다 보아 특징을 파악하는 것도 하나의 방법이라는 것을 배웠다.

4. 데이터의 수가 많이 부족하다고 생각되어 추가적으로 데이터를 수집하거나 생성이 필요하다고 느껴졌다. 특히 Price 를 예측하는 분석에서 Pirce 변수에 결측치가 존재하여 데이터 손실이 불가피하게 일어났다는 점에서 아쉬움을 느꼈다.

5. 데이터를 도메인 지식을 활용하거나 비즈니스 관점에서 다뤄본 적이 없었는데 이번 기회에 비즈니스 적 통계분석에 대해 배울 수 있어서 좋은 경험이었다고 생각된다.

실습 과정을 통해 배운 또는 느낀 통찰, 아이디어, 애로사항 등을 정리합니다

한계점

1. 범주형 자료를 분석하는 것에 대해 어려움을 많이 겪었다.
   One-Hot encoding 을 통해 데이터를 변환하여 이를 해석하는 것에 대해 어려움을 겪었고, 변수들의 증가로 인해 수치형 변수의 수에 비해 범주형 변수를 encoding 한 변수들의 수가 많아져 수치형 변수들의 영향력을 가려버리는 효과가 나타나 수치형 변수들이 유의미한지 아닌지 파악할 수 없게 되었다.

   추후에 범주형 데이터를 One-Hot encoding 이 아닌 Label Encoding 혹은 PCA 기법을 적용하여 차원의 수를 줄이는 공부를 해야겠다고 생각된다.

2. 선형회귀분석 도중 잔차의 독립성과 정규성을 제대로 이루어 내지 못했다고 생각된다. 이는 범주형 변수 간의 상관성 혹은 범주형 변수와 수치형 변수간의 상관성으로 인해 생긴 문제점이라고 생각된다. 따라서 이 둘간의 상관성과 처리방법을 추후에 공부할 필요가 있다고 생각된다.

3. 짧은 분석기간 내에 많은 분석을 하여 GridSearchCV나 RandomSearchCV를 이용하여 다양한 Hyperparameter 들을 탐색해보지 못했다. 조금 더 많은 시도를 통해 설명력을 끌어올리지 못했다는 것에 아쉬움을 느낀다.

핵심인자 정리 템플릿

핵심인자 선정을 위한 분석 과정에서 나온 결과를 순위 등으로 종합 정리합니다.
각자 필요한 형식으로 변경해서 사용하세요(엑셀 파일 제공)

변 수	변수 설명	변수 역할	변수 형태	분석제외 사유	탐색적 기법			모델링 기법						총점	선정 (사유)
					그래프	t/F 검정	X2 검정	회귀 분석	DT	RF	GB	...	기술적 검토		
Price	중고차 가격(단위:천원)	목표변수	연속형												
Brand	자동차 브랜드와 모델	설명변수	연속형		o		o	9	11	12	11			43	o
Location	자동차를 팔거나 구매할 수 있는 위치	설명변수	범주형		o		o	2	12	10	10			34	o
Year	모델의 년도 혹은 버전	설명변수	연속형		o		o	11	-	-	-			11	
Kilometers_ Driven	이전 소유주의 차량 주행거리(Km)	설명변수	연속형		o		o	3	-	-	-			3	
Fuel_Type	자동차 사용연료	설명변수	범주형		o		o	4	10	11	9			35	o
Transmission	자동차 변속기 종류	설명변수	범주형		o		o	7	-	-				7	
Owner_Type	소유권이 직접 소유인지, 중고 소유인지 여부	설명변수	범주형		o		o	6	9	9	8			34	o
Mileage	자동차 회사가 제공하는 표준 주행거리(kmpl)	설명변수	연속형		o		o	8	-	-				8	
Engine	엔진의 배기량(cc)	설명변수	연속형	Power 와 높은 상관관계	x		-	-	-	-	-	-	-	-	
Power	엔진의 최대 출력(bhp)	설명변수	연속형		o		o	10	-	-				10	
Seats	차의 좌석 수	설명변수	연속형	독립성 위배	o		x	-	-	-	-	-	-	-	
New_Price	뉴모델의 가격	설명변수	연속형		o		o	6	-	-	12			18	o