

# 중고차 가격 예측 챗봇



차차조(김도연, 박노현, 박혜진, 이성수, 장윤주, 홍미미)

---

# 0 조원 소개 😊



# Contents

1

## 프로젝트 배경 및 목표

- 프로젝트 배경
- 프로젝트 목표

2

## 프로젝트 Process

- Data 수집 및 pre-processing
- Data Visualization
- Machine Learning 알고리즘
- 알고리즘 점수 비교

3

## 프로젝트 Result

- 최종 알고리즘
- 챗봇 구현

4

## 프로젝트 의의 및 보완점

- 프로젝트 의의
- 프로젝트 보완점

# Chapter 1

## 프로젝트 배경 및 목표



# 1 프로젝트 배경

주제 선정 배경

가격 역전 현상

코로나 19로 인한 부품 수급  
난으로 중고차의 가격이 새  
차의 가격을 넘는 가격 역전  
현상 발생

## 새차보다 비싼 중고차... 車공급난에 '1년 안된 SUV' 등 수요 급증

변종국 기자 입력 2021-08-25 03:00 수정 2021-08-25 04:09

차량용 반도체 등 부품난 이어져

완성차 업계 차량 생산량 조절...구매후 신차 받는데 반년이상 걸려  
“당장 탈수있는 중고차가 낫다”, ‘출고 얼마 안된 신차급’ 웃돈 줘야  
새차보다 수백만원 비싸게 팔려...미국에선 40% 가까이 오르기도

### 8월 중고차 매입 시세 상승률

단위: %, 7월 대비 상승률.



자료: AJ 셀카, 첫차



# 1 프로젝트 배경

## 주제 선정 배경

### 중고차 거래 250만대 넘었다… 25% 훌쩍

최근 5년래 최대 규모… 대중교통 기피에 코로나 비껴가  
사업자 매입·매도 나란히 늘어 '예상 밖 선전'  
성장에도 '레몬마켓' 오명 벗지 못해, 대기업 진출은 답보

박상재 기자 입력 2021-01-07 11:33 | 수정 2021-01-07 13:08



### [친절한 경제] '테슬라 표' 중고차 나온다…우리나라는?

김혜민 기자 khm@sbs.co.kr 작성 2021.09.01 09:50 조회 3,488

▲ 나란히 주차된 자동차, 본 기사 내용



## 중고차 시장의 성장

중고차

258만대

190만대

신차

\* 2020년 기준

지난해 국내 중고차 거래 대수: 258만대, 매출액: 약 10조 원  
같은 기간 대비 국내 판매 신차: 190만대  
또한 미국의 테슬라 등 기업은 중고차 산업에 뛰어 들고 있음.

[https://news.sbs.co.kr/news/endPage.do?news\\_id=N1006453472](https://news.sbs.co.kr/news/endPage.do?news_id=N1006453472)

# 1 프로젝트 배경

현황 분석 - 소비자는 중고차 가격을 어떻게 알아볼까?



대한민국 No.1 직영중고차



중고차 가격 어떻게 알아볼까?

100만 회원 'KB차차차' 통해 합리적 중고차 시세 제공

[오토 파이낸스] KB캐피탈

김우영 기자

입력 2021.09.17 03:00



KB캐피탈 소속 진단 전문가가 중고차 매매단지를 방문해 판매 차량을 살펴보고 있다. / KB캐피탈 제공

FOI  
한화 포

집을

10월



케이카, 국내 중고차 플랫폼 1위...온라인 경쟁력 '주목' - 유안타

등록 2021-09-28 오전 7:41:18  
수정 2021-09-28 오전 7:41:18

가 가



김재은 기자

N 7/24/2021



☆ 스크랩

URL 복사

지금 열독 중

케이카 IPO 일정

자료:케이카

수요 예측 9월 27~28일

청약 일정 9월 30~10월 1일

코스피 상장 10월 중

공모 주식수 1683만 228주

주당 공모가액 3만 4300~4만 3200원

공모 예정금액 5773억~7271억원

예상 시가총액 1조 7454억~2조 1983억원

그래픽:이데일리 문송용 기자

대한민국 No.1  
롯데렌터카

바이드림카



# 1 프로젝트 목표

---

**A**



**B**



**C**





## Chapter 2

### 프로젝트 Process

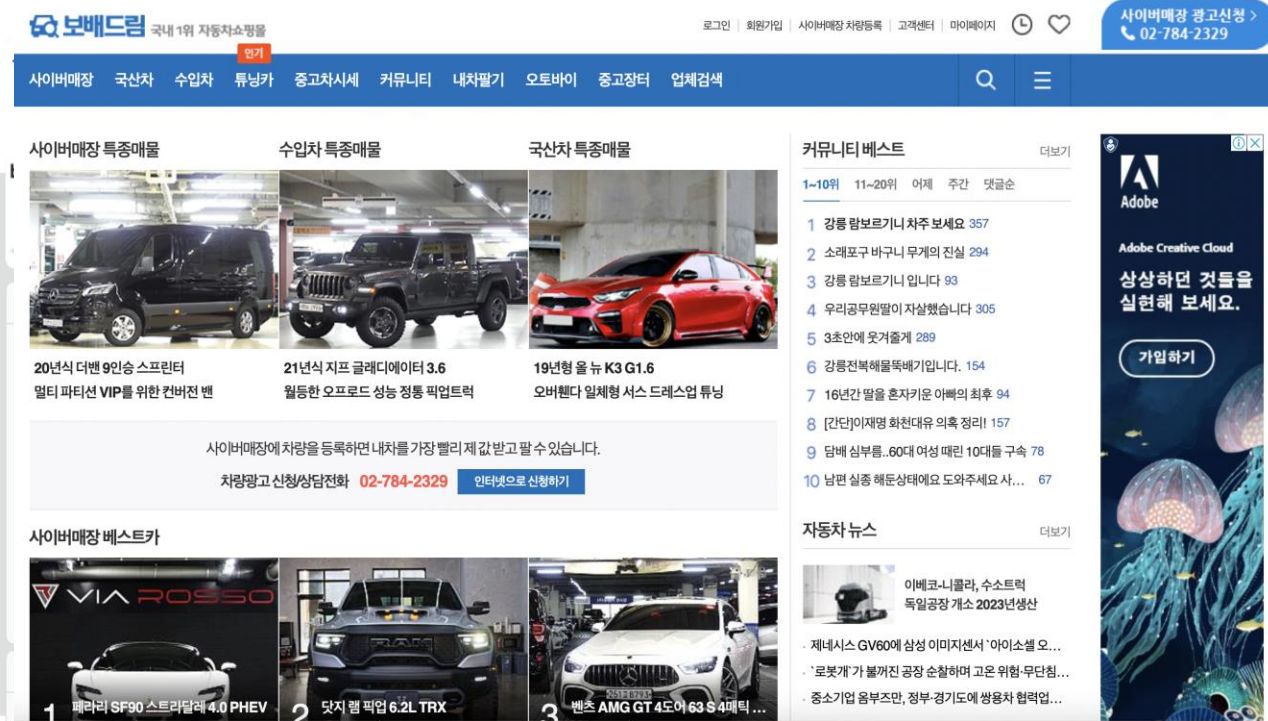


# 2 프로젝트 Process

## Data 수집



The image shows the KB CarChaCha app interface. At the top, there's a navigation bar with 'KB차차차' logo and search, status, price, market, benefits, and introduction tabs. Below this, a large banner features a man holding a smartphone displaying the app, with a line graph showing a downward trend in values (2,426, 2,343, 2,182, 2,087). The text '차량관리는 다 알아서 행해주는 KB차차차 내차고' is prominent. Below the banner, there's a section for '내차고' (My Car) with icons for license/registration, inspection, transfer, and market price. A sidebar on the right lists various services like '이벤트', '새소식', '전체서비스', '회원가입', and '로그인'.



The image shows the Bobaedream website interface. At the top, there's a navigation bar with '보배드림' logo and search, login, and account tabs. Below this, there's a section for '사이버매장' (Cyber Store) with sub-sections for '특종매물' (Special Offer), '수입차' (Import Car), and '국산차' (Domestic Car). The '특종매물' section lists various cars with their specifications and prices. The '수입차' section lists cars with their specifications and prices. The '국산차' section lists cars with their specifications and prices. Below this, there's a section for '사이버매장 베스트카' (Cyber Store Best Car) with three cars: '페라리 SF90 스트라달레 4.0 PHEV', '닷지 램 픽업 6.2L TRX', and '벤츠 AMG GT 4도어 63 S 4메트릭...'. On the right, there's a section for '커뮤니티 베스트' (Community Best) with a list of posts. At the bottom, there's a section for '자동차 뉴스' (Car News) with a list of news items.

1

KB 차차차

2

보배드림

## 2 프로젝트 Process

Data 수집

제조사

국산차

현대

기아

한국GM

르노삼성

쌍용

제네시스

기타

차종



경차



소형차



준중형차



중형차



대형차



SUV



RV



스포츠카



승합차



트럭/화물

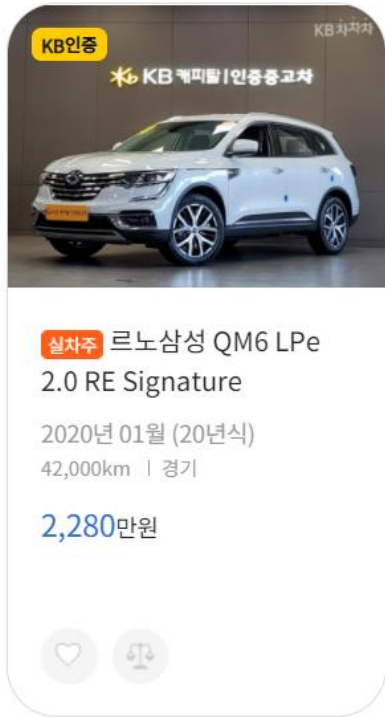
\* 국산차 기준

· 보배드림 : 16949개, KB차차차: 52100개

· 총 수집 데이터: 69049개

## 2 프로젝트 Process

### Data 수집



#### 기본정보

차량정보	140하1845	연식	20년01월 (20년형)
주행거리	42,000km	연료	LPG
변속기	CVT	연비	8.6Km
차종	SUV	배기량	1,998cc
색상	흰색	세금미납	없음
압류	없음	저당	없음
제시번호	20210910		

항목	내용
연비	자동차가 1리터당 주행할 수 있는 거리 연비가 높으면 동일한 양의 기름을 넣어도 더 많은 거리 주행 가능
배기량	엔진의 크기. 엔진이 크면 출력이 높아져 마력과 토크 가 좋아짐 1600cc미만 : 소형 1600cc~2000cc : 중형 2000cc 이상: 대형
변속기	각종 엔진에서 발생하는 동력을 속도에 따라 필요한 회 전력으로 바꾸어 전달하는 변속장치 수동(Manual)방식과 자동(Auto)방식으로 나눔

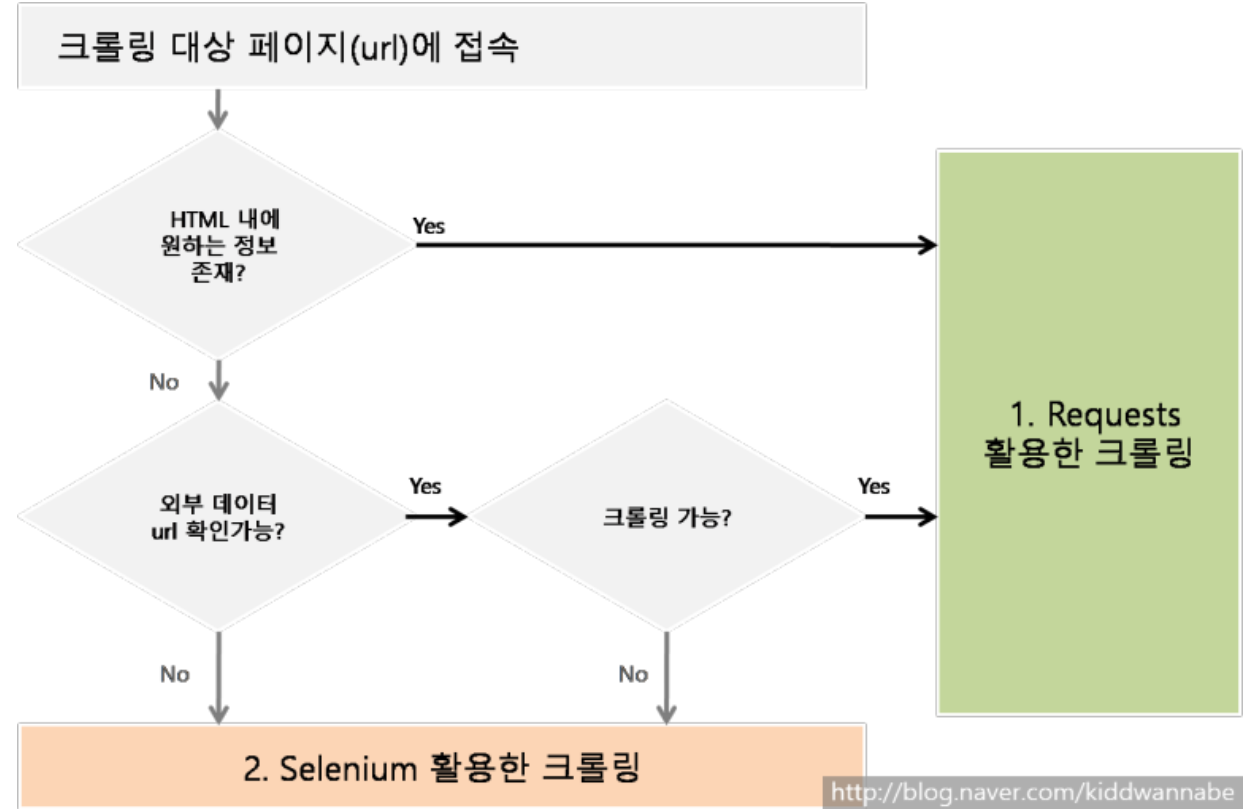
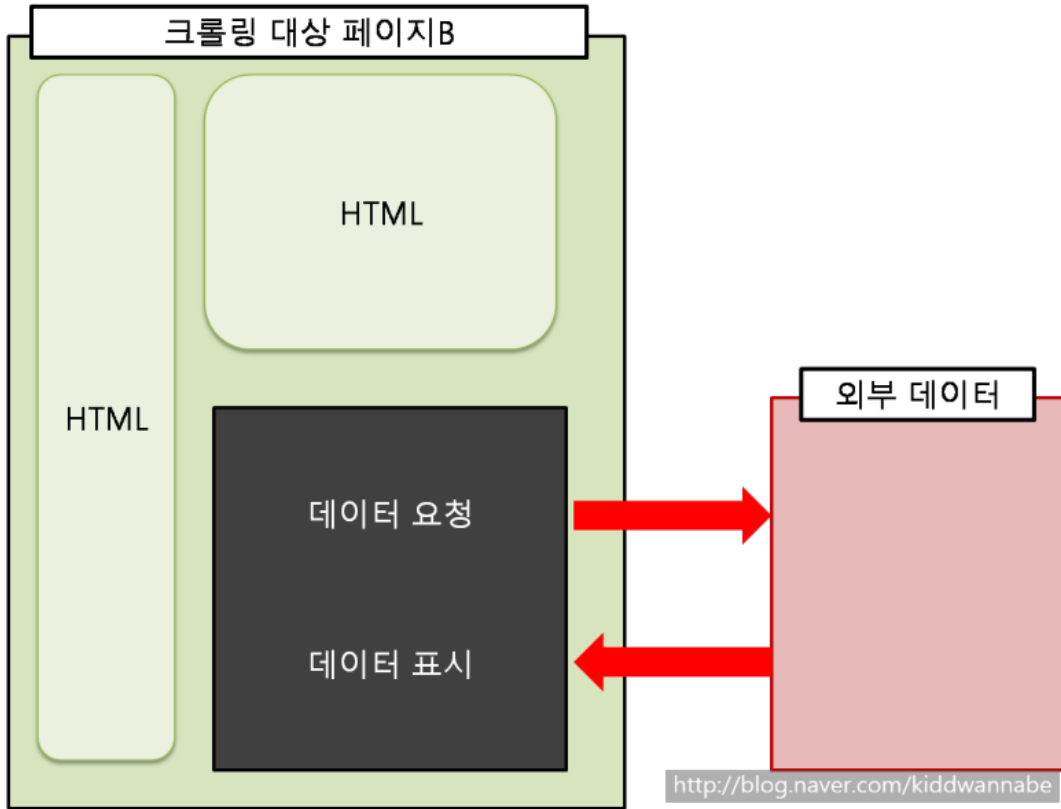
예시. KB 차차차



[이름, 차종, 가격, 연식, 주행거리, 연료, 변속기, 연비, 배기량] 수집

## 2 프로젝트 Process

Data 수집 - Selenium 패키지



해당 사이트는 Java Script를 통해 서버에 데이터를 요청 후 홈페이지에 불러오는 형식  
HTML 정보가 없는 부분은 기존의 방식으로 크롤링 불가능 → Selenium 방식을 통해 직접 서버에 데이터 요청하여 크롤링



# 2 프로젝트 Process

Data 수집







외부링크

[차종, 이름, 연식, 주행거리, 가격] 수집

내부링크

[배기량, 변속기, 연료, 색상] 수집

← → ↻ [bobaedream.co.kr/mycar/mycar\\_list.php?gubun=K](http://bobaedream.co.kr/mycar/mycar_list.php?gubun=K)

시각-동영상	차량정보	연식	연료	주행	가격	지역 / 판매자	최근본차량
	현대 올 뉴 아반떼 1.6 모던 자동 · 5인승 · 4기통 · 123마력 · 15.7kgm · FF · 선택가능 · 보험이력	20/09 (21년형)	가솔린	1만km	2,070 만원	박찬빈 (딜러) 경기 의정부시 등록 09/30 조회 240	1/1 비교하기 0 원함차량 비교검색 TOP
	현대 아반떼MD M16 GDi 프리미어 자동 · 5인승 · 140마력 · 17.0kgm · FF · 선택가능 · 보험이력	11/03	가솔린	11만km	680 만원	박찬빈 (딜러) 경기 의정부시 등록 09/30 조회 124	
	현대 아반떼AD 1.6 e-VGT 스마트 자동 · 5인승 · 4기통 · 136마력 · 26.9kgm · FF · 선택가능 · 보험이력	18/03	다젤	4만km	1,420 만원	박찬빈 (딜러) 경기 의정부시 등록 09/30 조회 171	
	기아 더 뉴 레이 1.0 벤 럭셔리 자동 · 선택가능 · 보험이력	20/06	가솔린	1만km	1,290 만원	박찬빈 (딜러) 경기 의정부시 등록 09/30 조회 265	
	기아 올 뉴 카니발 2.2 디젤 9인승 럭셔리 자동 · 9인승 · 4기통 · 202마력 · 45kgm · FF · 선택가능 · 보험이력	17/08 (18년형)	다젤	11만km	1,650 만원	박찬빈 (딜러) 경기 의정부시 등록 09/30 조회 321	
	현대 테뉴 G1.6 플렉스 자동 · 5인승 · 4기통 · 123마력 · 15.7kgm · FF · 선택가능 · 보험이력	20/09	가솔린	7천km	1,990 만원	박찬빈 (딜러) 경기 의정부시 등록 09/30 조회 607	

← → ↻ [bobaedream.co.kr/mycar/mycar\\_view.php?no=2132781&gubun=K](http://bobaedream.co.kr/mycar/mycar_view.php?no=2132781&gubun=K)

기본정보

연식	2012.01	배기량	998 cc (82마력)
주행거리	103,508 km	색상	진주색
변속기	자동	보증정보	만료
연료	가솔린	확인사항	<a href="#">차량등록증</a> <a href="#">사원증</a>

시세정보



동급매물

팔린매물

중고시세

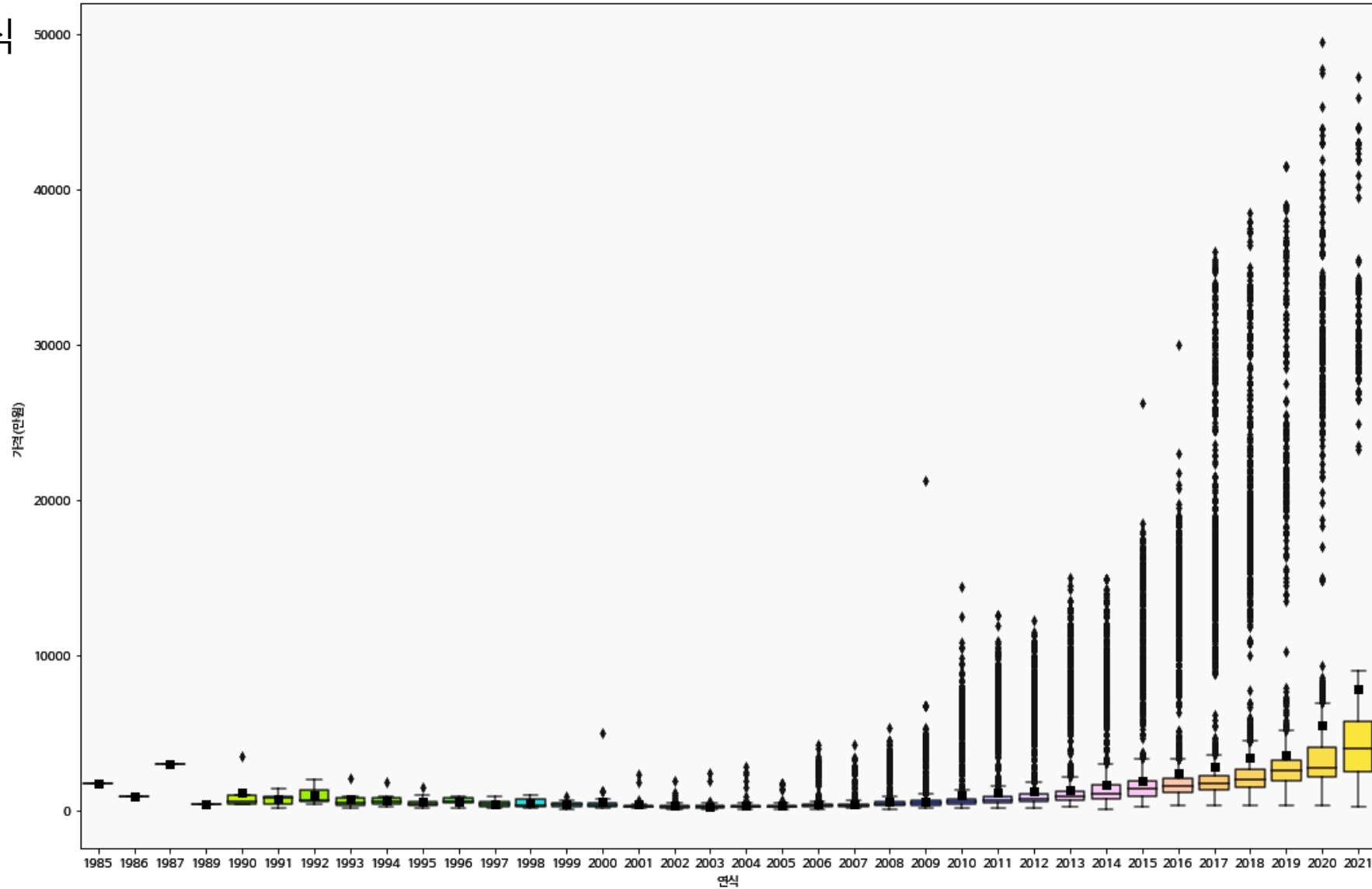


**Data visualization**

## 2 프로젝트 Process

Data visualization

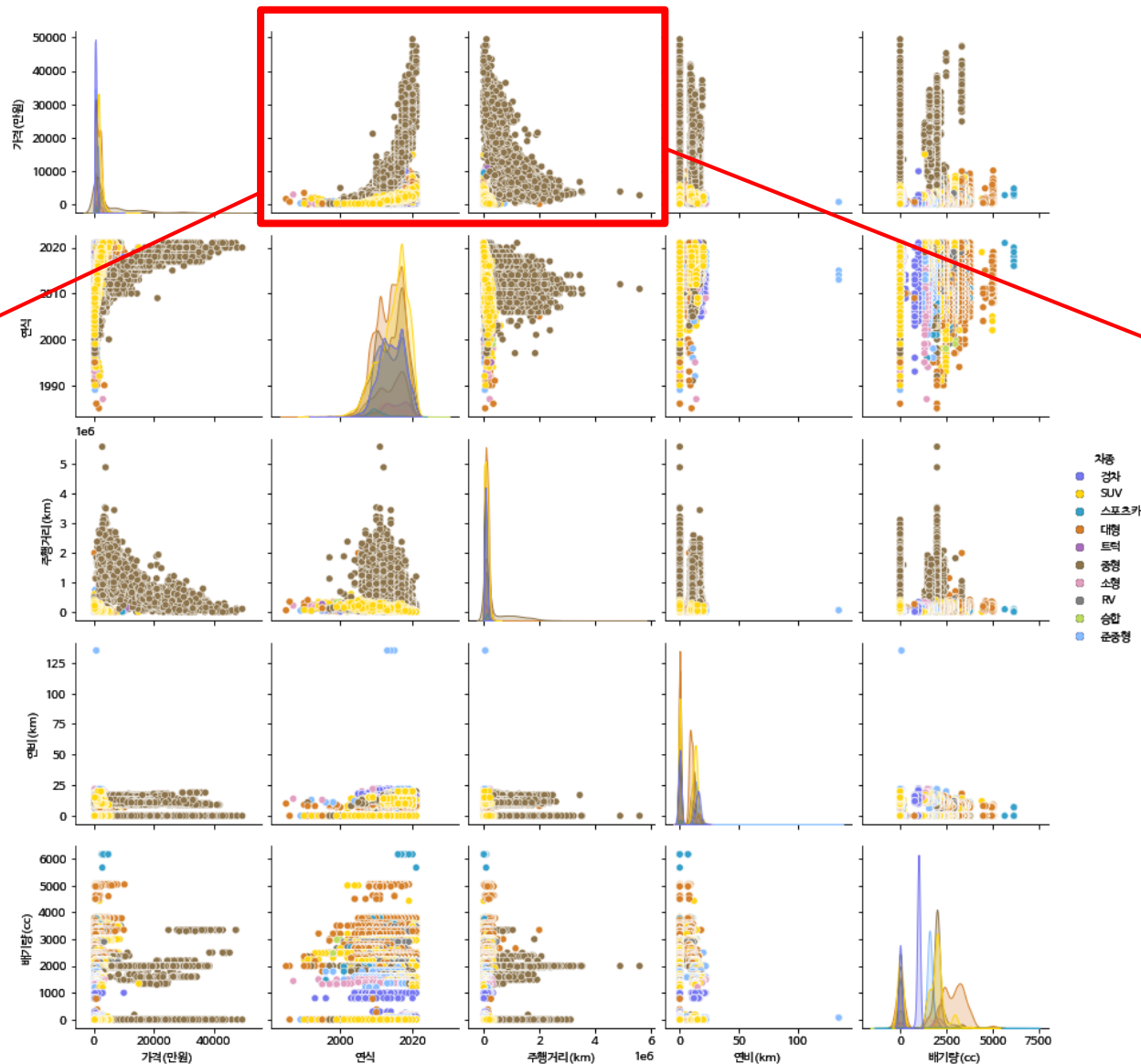
- 가격 ~ 연식



# 2 프로젝트 Process

Data visualization

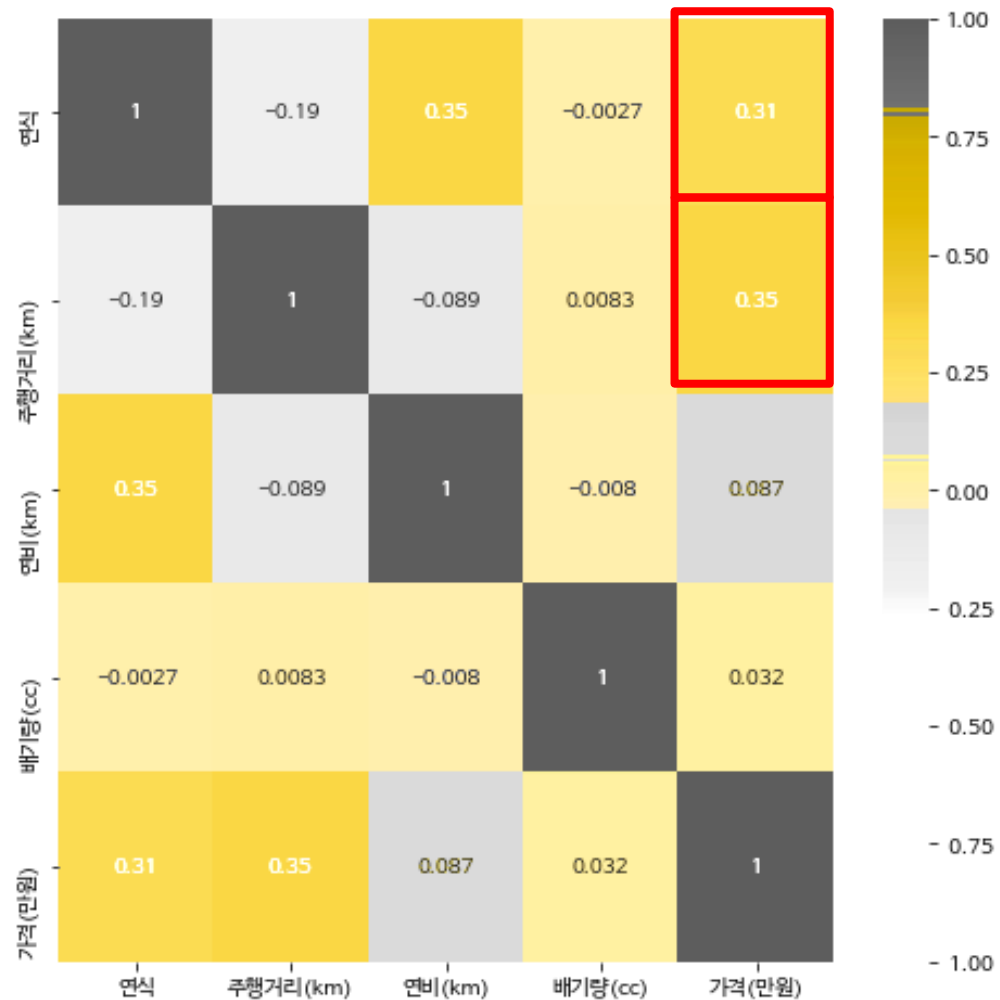
## ● Pair Plot



## 2 프로젝트 Process

Data visualization

- Pearson Correlation Coefficient (피어슨 상관계수)





## 2 프로젝트 Process

### Data pre-processing

	제조사	모델	차종	가격(만원)	연식	주행거리(km)	연료	변속기	연비(km)	배기량(cc)
0	기아	모닝	경차	210	2008	100000	가솔린	오토	0.0	991.0
1	쌍용	티볼리	SUV	1860	2019	4427	가솔린	오토	12.0	4427.0
2	현대	제네시스 쿠페	스포츠카	870	2010	142395	가솔린	오토	0.0	3778.9
3	제네시스	BH330	대형	990	2009	171709	가솔린	오토	0.0	3300.0
4	현대	제네시스	대형	990	2009	171709	가솔린	오토	0.0	3300.0
...	...	...	...	...	...	...	...	...	...	...
65871	기아	스포티지	SUV	610	2009	134118	가솔린	오토	0.0	0.0
65872	기아	스포티지	SUV	250	2006	290000	가솔린	오토	0.0	0.0
65873	기아	스포티지	SUV	300	2009	238700	디젤	오토	15.0	0.0
65874	기아	스포티지	SUV	350	2006	119506	디젤	오토	0.0	0.0
65875	기아	스포티지	SUV	249	2005	211505	디젤	오토	0.0	0.0

65876 rows × 10 columns

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 65876 entries, 0 to 65875
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
---  -
0   제조사      65876 non-null object
1   모델        65876 non-null object
2   차종        65876 non-null object
3   가격(만원)  65876 non-null int64
4   연식        65876 non-null int64
5   주행거리(km) 65876 non-null int64
6   연료        65876 non-null object
7   변속기      65876 non-null object
8   연비(km)    65876 non-null float64
9   배기량(cc)  65876 non-null float64
dtypes: float64(2), int64(3), object(5)
memory usage: 5.0+ MB
```

- Null 값 제외
- 이름 → 제조사/모델 구분하여 컬럼 생성
- 연식 (yyyy년mm월 → yyyy)
- 데이터 타입 변환
- 색상 컬럼 제거

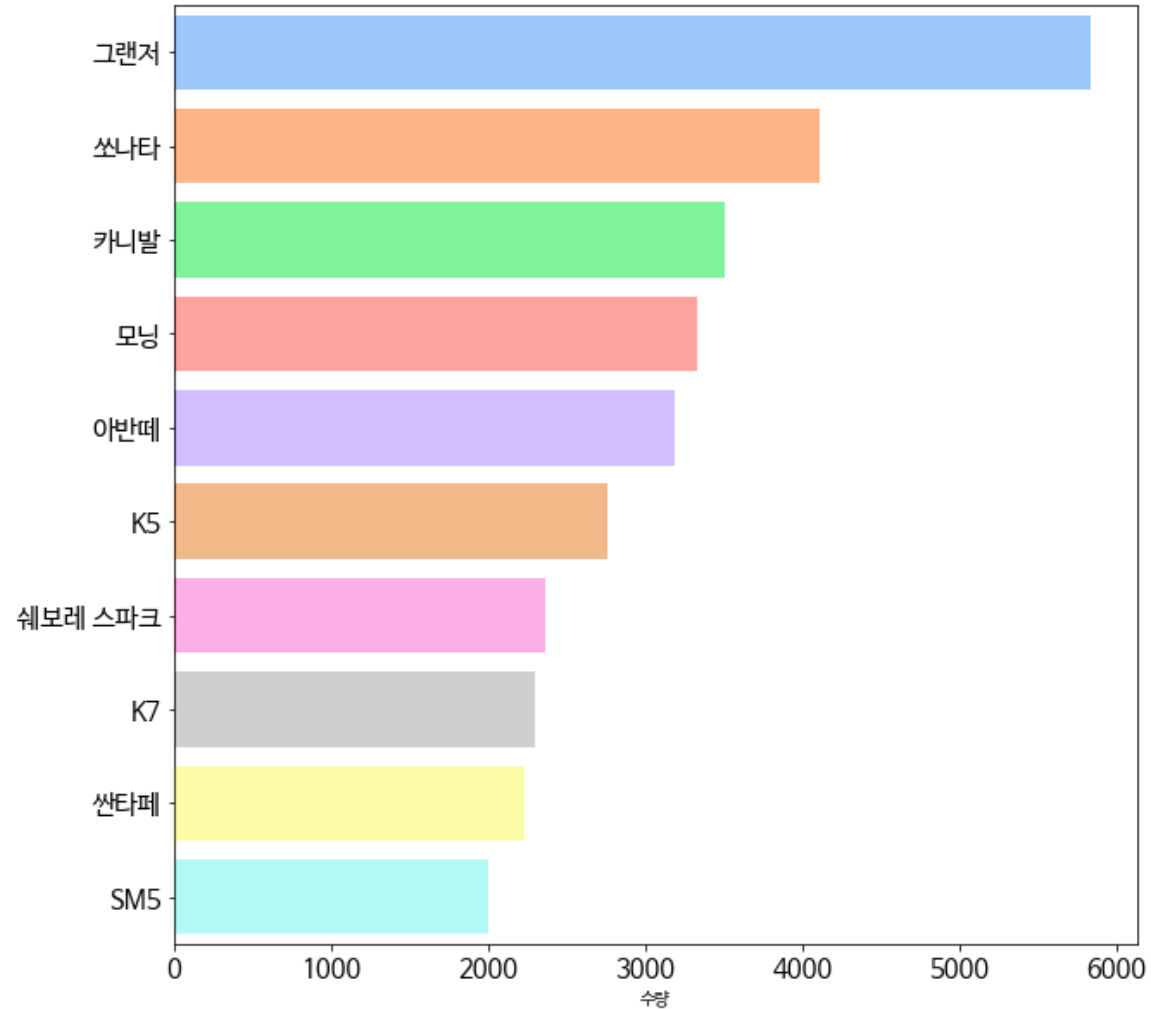


전처리 후 총 데이터 65876개

## 2 프로젝트 Process

Data visualization

- 차량 모델 매물 수량 Top10



A large yellow rectangle is positioned on the left side of the slide. Overlapping its right edge is a smaller, semi-transparent gray rectangle. The text 'Machine Learning Algorithm' is centered within the gray rectangle.

# Machine Learning Algorithm

# 2 프로젝트 Process

Machine Learning Algorithm

---

Linear  
Regression

Logistic  
Regression

SGD

Ridge

Gradient  
boosting

Random  
Forest

## 2 프로젝트 Process

Data pre-processing

	Train score	Test score
Linear Regression	0.27	0.27
Logistic Regression	0.297	0.311
SGD	0.26	0.28
Ridge	0.26	0.28
Gradient Boosting	0.79	0.76
Random Forest	0.97	0.82

- 전체 데이터를 6개 알고리즘에 적용하여 실행하였을 때,  
Random Forest를 제외한 알고리즘 결과 값이 매우 낮게 나와 추가 전처리가 필요하다고 판단



## 2 프로젝트 Process

- Encoding

ID	과일
1	사과
2	바나나
3	체리

One-Hot Encoding

ID	사과	바나나	체리
1	1	0	0
2	0	1	0
3	0	0	1

LabelEncoder

ID	과일
1	0
2	1
3	2

단 하나의 값만 True(1)이고, 나머지는 모두 False(0)  
데이터 형태가 0과 1로 이루어졌기 때문에  
컴퓨터가 인식하고 학습하기에 용이

인코딩된 숫자를 가중치로 인식하여  
예측 값에 영향을 미칠 수 있음

## 2 프로젝트 Process

- One-Hot Encoding 실행 (제조사, 모델, 차종, 연료, 변속기 컬럼) → DataFrame의 컬럼 수 총 193개

```
1 one_hot = pd.get_dummies(df)
2 one_hot
```

	가격 (만원)	연식	주행거리(km)	연비(km)	배기량(cc)	제 조 사_ 기 아	제 조 사_ 기 타	제 조 사_ 르 노 삼 성	제 조 사_ 쌍 용	제 조 사_ 쌍 용	제 조 사_ 제 네 시스	제 조 사_ 한 국 GM	제 조 사_ 현 대	모 델_ 쉐 보 레 말 리 부	모 델 _3.8	모 델 _BH330	모 델 _BH380	모 델 _BH460	모 델 _EQ900	모 델 _G2X	모 델 _G330	모 델 _G380	모 델 _G70	모 델 _G80	모 델 _G90
0	210	2008	100000	0.0	991.0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	1860	2019	4427	12.0	4427.0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	870	2010	142395	0.0	3778.9	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
3	990	2009	171709	0.0	3300.0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	
4	990	2009	171709	0.0	3300.0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
65871	610	2009	134118	0.0	0.0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
65872	250	2006	290000	0.0	0.0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
65873	300	2009	238700	15.0	0.0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
65874	350	2006	119506	0.0	0.0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
65875	249	2005	211505	0.0	0.0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

65876 rows × 193 columns

# 2 프로젝트 Process

## Machine Learning Algorithm

- Linear Regression

```
model_lig = Pipeline(steps = [('scaler', StandardScaler()),  
                              ('lin_reg', LinearRegression())])
```

```
model_lig.fit(X_train, y_train)  
> Pipeline(memory=None,  
            steps=[('scaler',  
                    StandardScaler(copy=True, with_mean=True, with_std=True)),  
                  ('lin_reg',  
                    LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,  
                                      normalize=False))],  
            verbose=False)  
train_pred = model_lig.predict(X_train)
```

```
r2_score(y_train, train_pred)  
> 0.785302385256714
```

Train score

0.79

```
test_pred = model_lig.predict(X_test)
```

```
r2_score(y_test, test_pred)  
> -9.795710055822667e+23
```

Test score

- 9.79

# 2 프로젝트 Process

## Machine Learning Algorithm

- Logistic Regression

```
model_log = Pipeline(steps = [('scaler', StandardScaler()),  
                              ('log', LogisticRegression())])
```

```
model_log.fit(X_train, y_train)
```

```
train_pred = model_log.predict(X_train)
```

```
r2_score(y_train, train_pred)  
> 0.8594217454001662
```

Train score

0.86

```
test_pred = model_log.predict(X_test)
```

```
r2_score(y_test, test_pred)  
> 0.8377223824084964
```

Test score

0.84

# 2 프로젝트 Process

## Machine Learning Algorithm

- SGD(Stochastic Gradient Descent)

```
scaler = StandardScaler()  
reg = SGDRegressor()  
model = Pipeline(steps=[('scaler', scaler),  
                        ('reg', reg)])
```

```
model.fit(X_train, y_train)
```

```
train_pred = model.predict(X_train)
```

```
r2_score(y_train, train_pred)  
> -3599488263377243.0
```

Train score

매우 낮은  
값

```
test_pred = model.predict(X_test)
```

```
r2_score(y_test, test_pred)  
> -3218491120402272.0
```

Test score

매우 낮은  
값



# 2 프로젝트 Process

## Machine Learning Algorithm

- Ridge

```
scaler = StandardScaler()  
reg = Ridge(alpha=0.0001)  
model3 = Pipeline(steps=[('scaler', scaler),  
                           ('reg', reg)])
```

```
model3.fit(X_train, y_train)
```

```
train_pred = model3.predict(X_train  
)
```

```
r2_score(y_train, train_pred)  
> 0.7853634813567013
```

Train score

0.79

```
test_pred = model3.predict(X_test)
```

```
r2_score(y_test, test_pred)  
> 0.7681556580254789
```

Test score

0.77

# 2 프로젝트 Process

## Machine Learning Algorithm

- Gradient Boosting

```
grad_boost = GradientBoostingRegressor(random_state=42)
cv = cross_validate(estimator=grad_boost, X=X_train, y=y_train, n_jobs=-1,
                    return_train_score=True)
```

```
np.mean(cv['train_score'])
> 0.9329537061856916
```

Train score

0.93

```
np.mean(cv['test_score'])
> 0.9237792039226012
```

Test score

0.92

# 2 프로젝트 Process

## Machine Learning Algorithm

- Random Forest

```
logreg = RandomForestRegressor(oob_score=True, n_jobs=-1, random_state=42)  
logreg.fit(X_train, y_train)
```

```
logreg.score(X_train, y_train)  
> 0.9951870507258435
```

Train score

0.99

```
logreg.oob_score_  
> 0.9646312901793431
```

**Out of  
bagging score**

0.964

```
logreg.score(X_test, y_test)  
> 0.9667879132911736
```

Test score

0.966

## 2 프로젝트 Process

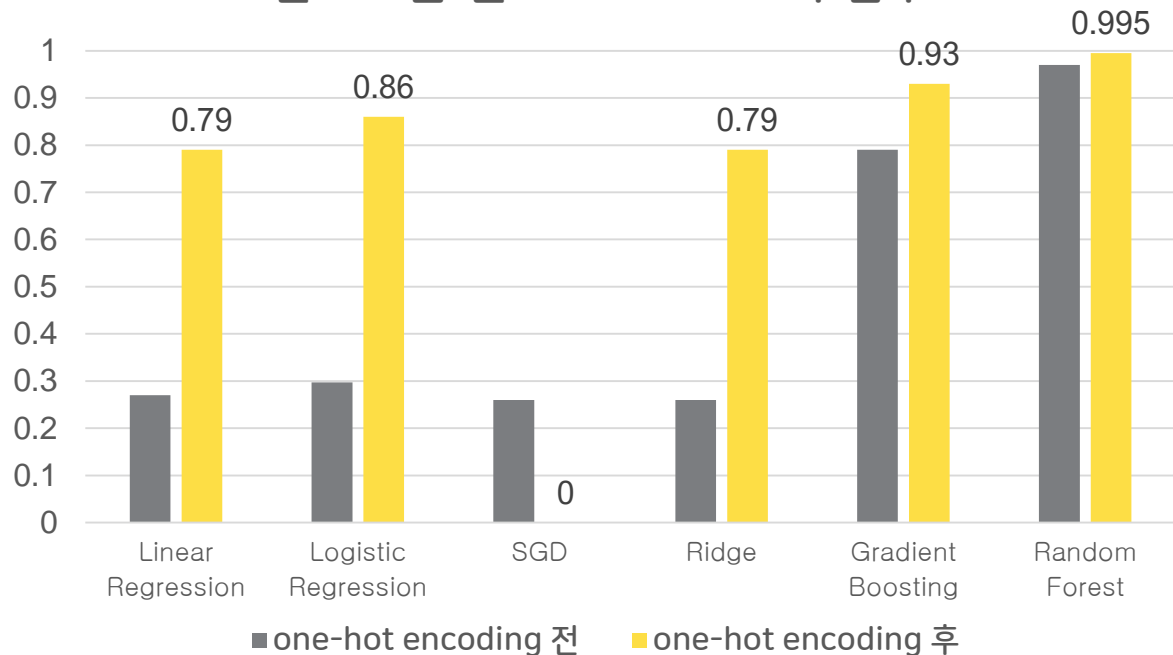
알고리즘 점수 비교

	Train score	Test score
Linear Regression	0.79	-9.80E+23
Logistic Regression	0.86	0.84
SGD	매우 낮은 값	매우 낮은 값
Ridge	0.79	0.77
Gradient Boosting	0.93	0.92
Random Forest	0.995	0.97

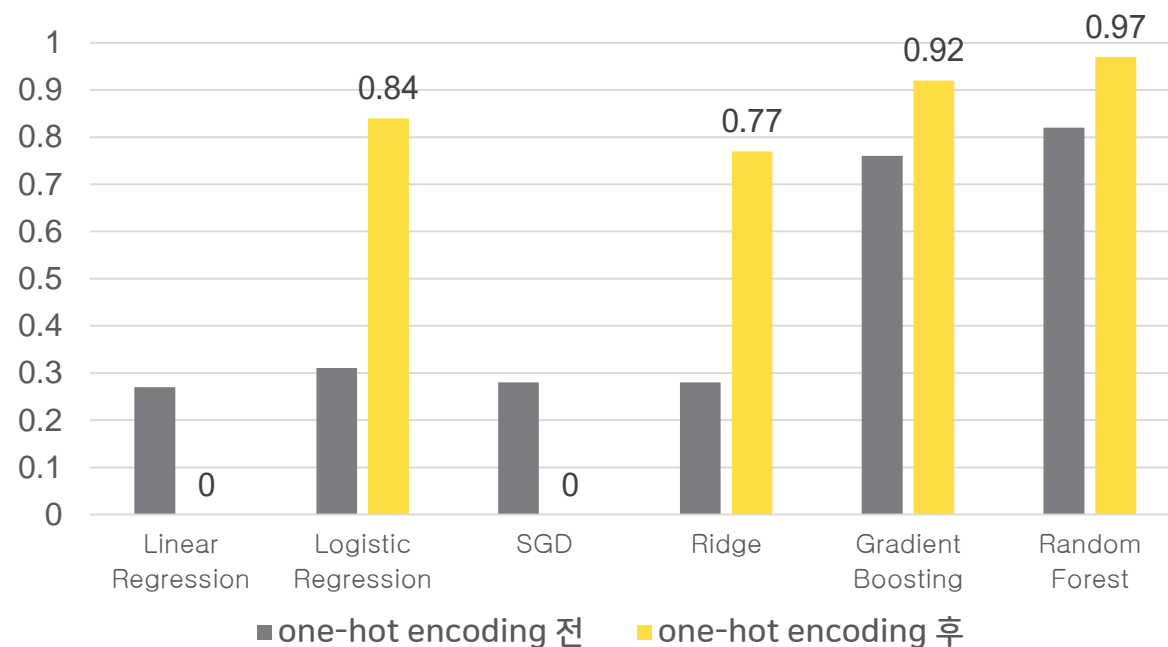
## 2 프로젝트 Process

알고리즘 점수 비교

알고리즘 별 Train set 예측 점수



알고리즘 별 Test set 예측 점수



## Chapter 3

### 프로젝트 Result



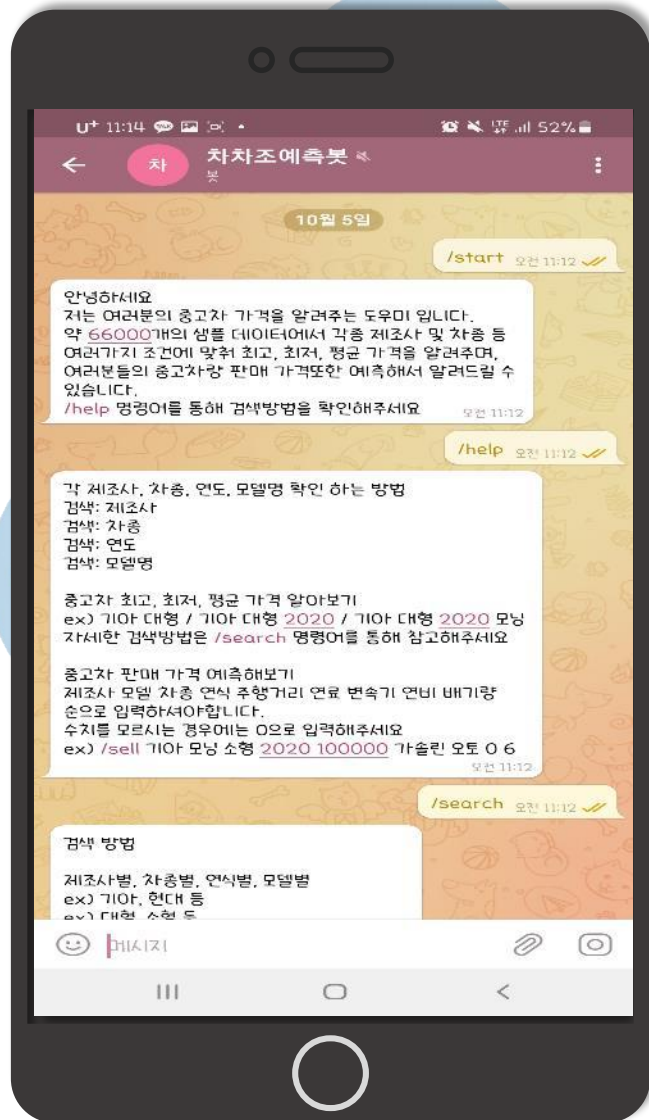
# Chatbot





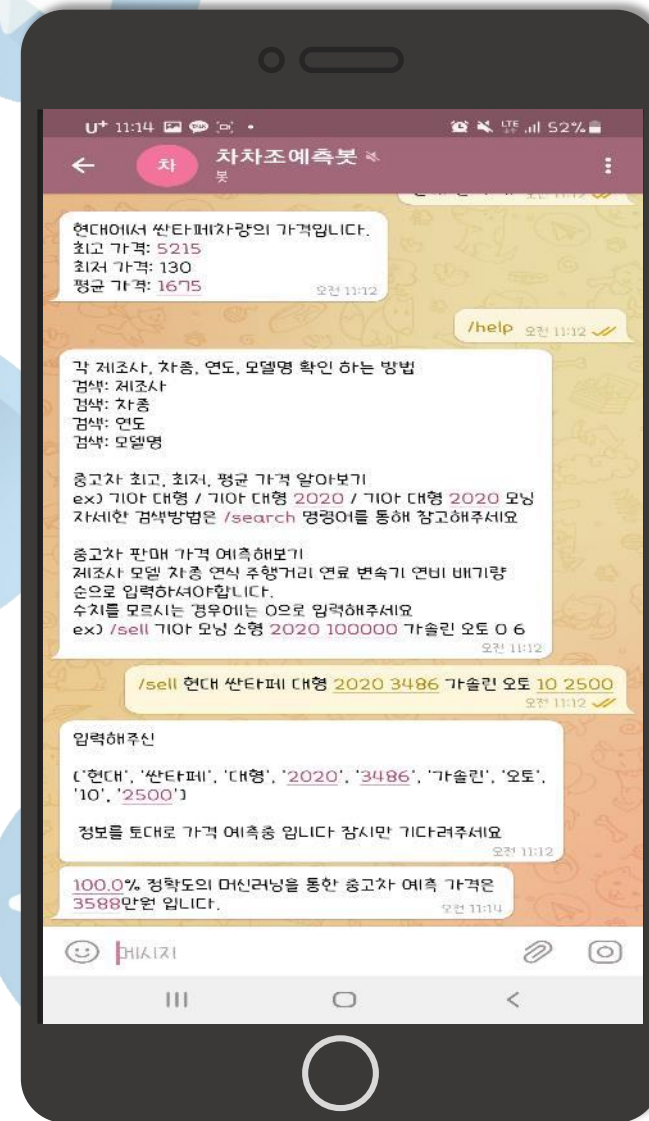
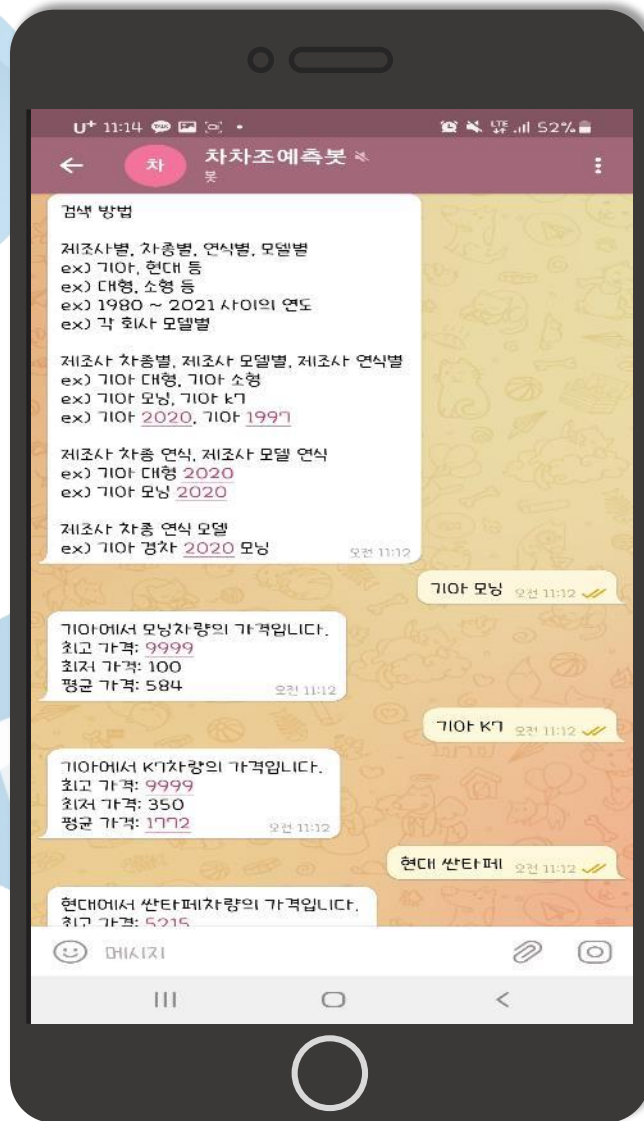
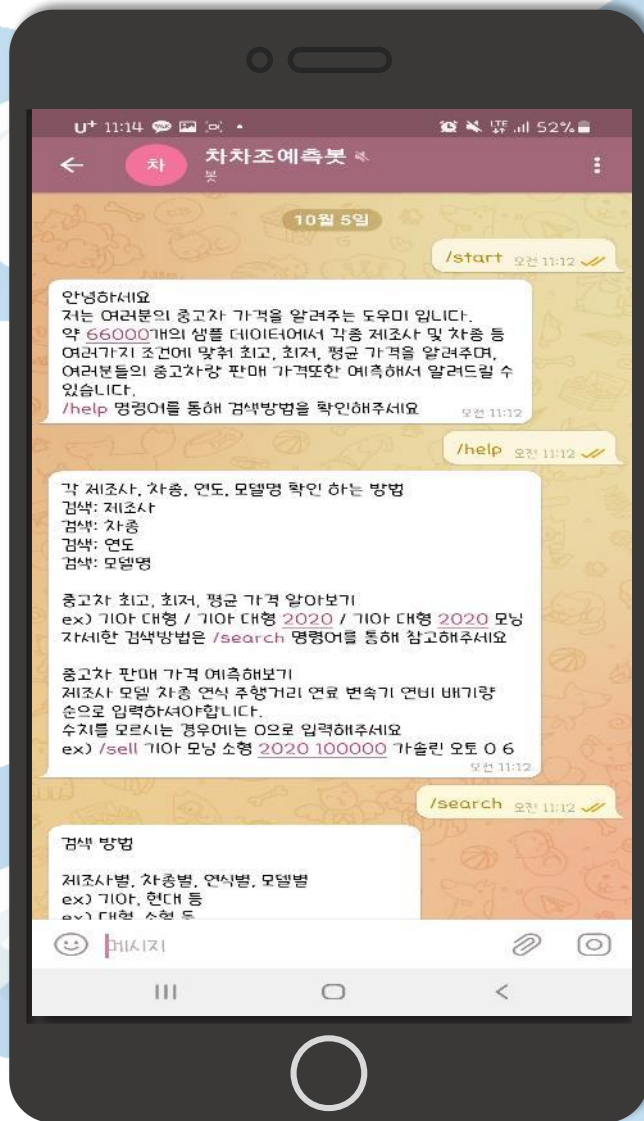
# 2 프로젝트 Result

입력형 챗봇 VS 버튼형 챗봇



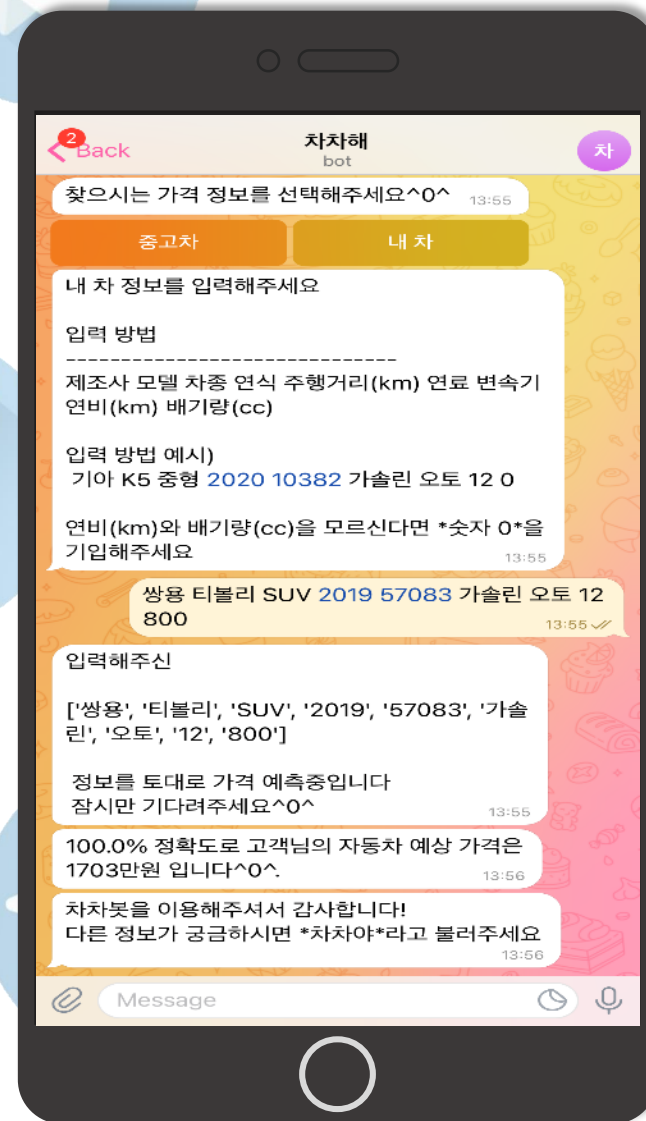
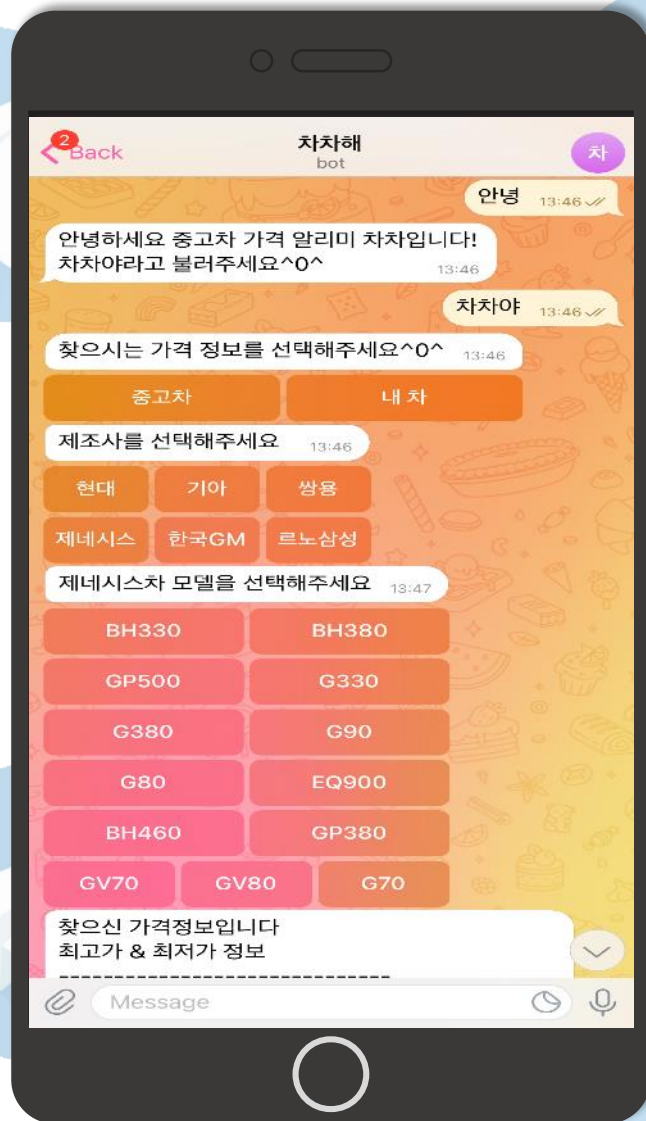
## 2 프로젝트 Result

입력형 챗봇 이미지 예시



## 2 프로젝트 Result

버튼형 챗봇 이미지 예시





## Chapter 4

프로젝트 의의 및 보완점



# 4 프로젝트 의의 및 보완점

의의 및 보완점



## 프로젝트 보완점

- 실시간으로 데이터 수집이 되지 않아 데이터의 주기적 업데이트 필요
- 보다 정확한 가격예측을 위한 수집 데이터 범위 확대 필요(보험, 블랙박스, 스마트키의 유무 등)
- 데이터 전처리 시 제조사와 모델명으로 분류해 세부 모델 별 가격차를 고려하지 못함



## 프로젝트 의의

- 가격을 결정하는 데 있어서 연식과 주행거리가 중요한 변수로 작용
- 알고리즘을 훈련 시킬 때 데이터 형태가 훈련 결과에 큰 영향을 미침
- 사용자들의 직관적 이해도를 고려하여 두 가지 형식의 챗봇으로 구현하여 편리성 증대

**Thank you !**

