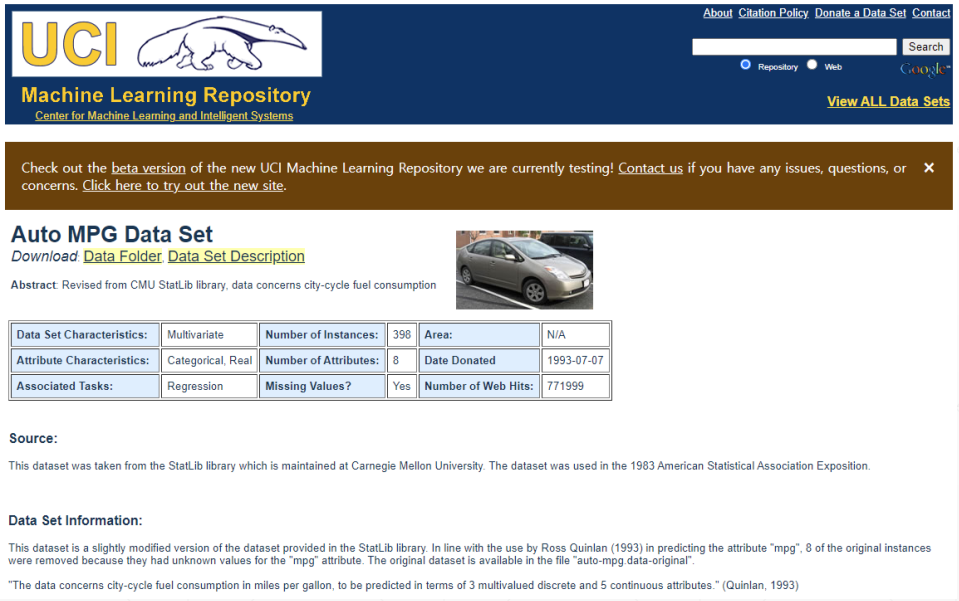
# 회귀 분석 II

빅데이터 분석

# 항목에 따른 자동차 연비 예측하기

**목표 :** 자동차 연비 데이터에 머신러닝 기반의 회귀 분석을 수행

**연비에 영향을 미치는 항목**을 확인하고, 그에 따른 **자동차 연비를 예측**




https://archive.ics.uci.edu/ml/datasets/auto+mpg

# 항목에 따른 자동차 연비 예측하기 (cont'd)

## 1. 데이터 준비 및 탐색

```
>>> import numpy as np
>>> import pandas as pd
>>> data_df = pd.read_csv('./auto-mpg.csv', header = 0, engine = 'python')

>>> print('데이터셋 크기: ', data_df.shape)
>>> data_df.head()

>>> data_df = data_df.drop(['car_name', 'origin', 'horsepower'], axis = 1,
inplace = False)
>>> data_df.head()

>>> print('데이터셋 크기: ', data_df.shape)

>>> data_df.info()
```

# 항목에 따른 자동차 연비 예측하기 (cont'd)

## 2. 선형 회귀 분석 모델 구축하기

```
>>> from sklearn.linear_model import LinearRegression
>>> from sklearn.model_selection import train_test_split
>>> from sklearn.metrics import mean_squared_error, r2_score

# X, Y 분할하기
>>> Y = data_df['mpg']
>>> X = data_df.drop(['mpg'], axis = 1, inplace = False)

# 훈련용 데이터와 평가용 데이터 분할하기
>>> X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3,
random_state = 0)

# 선형 회귀 분석 : 모델 생성
>>> lr = LinearRegression()

# 선형 회귀 분석 : 모델 훈련
>>> lr.fit(X_train, Y_train)
```

# 항목에 따른 자동차 연비 예측하기 (cont'd)

## 2. 선형 회귀 분석 모델 구축하기

```
# 선형 회귀 분석 : 평가 데이터에 대한 예측 수행 -> 예측 결과 Y_predict 구하기
>>> Y_predict = lr.predict(X_test)

# 평가
>>> mse = mean_squared_error(Y_test, Y_predict)
>>> rmse = np.sqrt(mse)
>>> print('MSE : {0:.3f}, RMSE : {1:.3f}'.format(mse, rmse))
>>> print('R^2(Variance score) : {0:.3f}'.format(r2_score(Y_test, Y_predict)))

>>> print('Y 절편 값: ', np.round(lr.intercept_, 2))
>>> print('회귀 계수 값: ', np.round(lr.coef_, 2))

>>> coef = pd.Series(data = np.round(lr.coef_, 2), index = X.columns)
>>> coef.sort_values(ascending = False)
```

# 항목에 따른 자동차 연비 예측하기 (cont'd)

### 3. 회귀 분석 결과를 산점도 + 선형 회귀 그래프로 시각화하기

```
>>> import matplotlib.pyplot as plt
>>> import seaborn as sns

>>> fig, axs = plt.subplots(figsize = (16, 16), ncols = 3, nrows = 2)
>>> x_features = ['model_year', 'acceleration', 'displacement', 'weight',
'cylinders']
>>> plot_color = ['r', 'b', 'y', 'g', 'r']
>>> for i, feature in enumerate(x_features):
      row = int(i/3)
      col = i%3
      sns.regplot(x = feature, y = 'mpg', data = data_df, ax = axs[row][col],
        color = plot_color[i])
```

# 항목에 따른 자동차 연비 예측하기 (cont'd)

## 4. 연비 예측

```
>>> print("연비를 예측하고 싶은 차의 정보를 입력해주세요.")
>>> cylinders_1 = int(input("cylinders : "))
>>> displacement_1 = int(input("displacement : "))
>>> weight_1 = int(input("weight : "))
>>> acceleration_1 = int(input("acceleration : "))
>>> model_year_1 = int(input("model_year : "))

>>> mpg_predict = lr.predict([[cylinders_1, displacement_1, weight_1,
acceleration_1 , model_year_1]])

>>> print("이 자동차의 예상 연비(MPG)는 %.2f입니다." %mpg_predict)
```