

## 나의 첫모델 만들기

- 데이터 다운로드 후, 머신러닝 모델 만들어보고 제출해 보기

## Kaggle 대회

- URL : <https://www.kaggle.com/>
- Competitions 선택하면 다양한 대회 확인 가능.
- 대회 주제 : Bike Sharing Demand
- <https://www.kaggle.com/c/bike-sharing-demand>

## 데이터 다운로드

## 데이터 다운로드하기

- 가. <https://www.kaggle.com/c/bike-sharing-demand> 링크를 선택하여 웹 사이트 접속합니다.
- 나. Data를 선택합니다.
- 다. train.csv, test.csv, sampleSubmission.csv를 다운로드 받습니다.
- 라. 다운로드 받은 csv와 주피터 노트북 또는 py 파일은 동일한 폴더에 위치시킵니다.



같은 폴더내에 데이터 파일(csv파일)과 python 노트북 파일을 위치시킵니다.

## Data Fields

필드명	설명
datetime	hourly date + timestamp
season	1 = spring, 2 = summer, 3 = fall, 4 = winter
holiday	whether the day is considered a holiday
workingday	whether the day is neither a weekend nor holiday
weather	1: Clear, Few clouds, Partly cloudy, Partly cloudy 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
temp	temperature in Celsius (온도)
atemp	"feels like" temperature in Celsius (체감온도)
humidity	relative humidity (습도)
windspeed	wind speed (바람속도)
casual	number of non-registered user rentals initiated (비가입자 사용유저)

필드명	설명
registered	number of registered user rentals initiated (가입자 사용유저)
count	number of total rentals (전체 렌탈 대수)

## 1-1 라이브러리 불러오기

```
In [1]: import pandas as pd
        from sklearn.linear_model import LinearRegression
```

## 1-2 데이터 셋 준비

- train 은 학습을 위한 데이터 셋
- test 은 예측을 위한 데이터 셋
- ../data/bike : 상위폴더의 (data/bike 폴더 경로), 내 컴퓨터의 데이터 경로 지정.
- parse\_dates = [컬럼명] : 해당 컬럼을 시간형 자료로 불러옴.

```
In [3]: train = pd.read_csv("../data/bike/train.csv", parse_dates=['datetime'])
        test = pd.read_csv("../data/bike/test.csv", parse_dates=['datetime'])
```

## 1-3 간단한 데이터 탐색

- 데이터 행과 열은? (shape)
- 데이터의 개수와 자료형? (info())
- 데이터의 컬럼명은 무엇일까? (columns)
- 몇행만 데이터를 확인해 보자. (head)

```
In [4]: print(train.shape)
        print(test.shape)
```

```
(10886, 12)
(6493, 9)
```

```
In [5]: print(train.info())
        print() # 한줄 공백
        print(test.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10886 entries, 0 to 10885
Data columns (total 12 columns):
#   Column      Non-Null Count  Dtype
---  -
0   datetime    10886 non-null  datetime64[ns]
1   season      10886 non-null  int64
2   holiday     10886 non-null  int64
3   workingday  10886 non-null  int64
4   weather     10886 non-null  int64
5   temp        10886 non-null  float64
6   atemp       10886 non-null  float64
7   humidity    10886 non-null  int64
8   windspeed   10886 non-null  float64
9   casual      10886 non-null  int64
10  registered  10886 non-null  int64
11  count       10886 non-null  int64
dtypes: datetime64[ns](1), float64(3), int64(8)
memory usage: 1020.7 KB
None
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6493 entries, 0 to 6492
Data columns (total 9 columns):
#   Column      Non-Null Count  Dtype
---  -
0   datetime    6493 non-null  datetime64[ns]
1   season      6493 non-null  int64
2   holiday     6493 non-null  int64
3   workingday  6493 non-null  int64
4   weather     6493 non-null  int64
5   temp        6493 non-null  float64
6   atemp       6493 non-null  float64
7   humidity    6493 non-null  int64
8   windspeed   6493 non-null  float64
dtypes: datetime64[ns](1), float64(3), int64(5)
memory usage: 456.7 KB
None
```

- 기본적으로 데이터 셋은 실수형(float)과 정수형(int)로 이루어져 있다.
- non-null : 결측치가 없음.

```
In [6]: print(train.columns)
print()
print(test.columns)

Index(['datetime', 'season', 'holiday', 'workingday', 'weather', 'temp',
       'atemp', 'humidity', 'windspeed', 'casual', 'registered', 'count'],
      dtype='object')

Index(['datetime', 'season', 'holiday', 'workingday', 'weather', 'temp',
       'atemp', 'humidity', 'windspeed'],
      dtype='object')
```

- 컬럼을 확인해 보니, test에는 count와 casual, registered가 없음.

## 1-4 초간단 머신러닝 모델을 만들어보자.

- 학습을 하게 되면 모델은 다음과 같은 선형 방정식 모델이 된다.
  - $y(\text{자전거렌탈대수}) = a1(\text{temp}) + a2(\text{atemp}) + b$

- 학습을 하는 모델은 a1, a2, b의 값을 구해준다.

## 일부 데이터 선택

- 학습과 예측을 위한 데이터 선택(X\_train, X\_test, y\_train)
- temp : 온도, atemp : 체감온도

```
In [7]: f_names = ['temp', 'atemp']
X_train = train[f_names] # 학습용 입력 데이터
X_test = test[f_names]   # 예측을 위한 입력 데이터
```

## 출력 데이터(예측할) 선택

```
In [8]: label_name = 'count' # 렌탈 대수 (통계 : 종속변수)
y_train = train[label_name] # 렌탈 대수 변수 값 선택
```

## 모델 만들기 및 예측 순서

- 모델을 생성한다. model = 모델명()
- 모델을 학습한다. model.fit(입력값, 출력값)
- 모델을 이용하여 예측 model.predict(입력값)

```
In [9]: # 모델 사용을 위한 준비(라이브러리 가져오기)
from sklearn.linear_model import LinearRegression
```

```
In [10]: # 모델 생성, 학습, 예측
model = LinearRegression()
model.fit(X_train, y_train)
model.predict(X_test) # 예측(새로운 데이터로)
```

```
Out[10]: array([101.95625474, 104.0156171 , 104.0156171 , ..., 103.33067499,
               104.0156171 , 104.0156171 ])
```

## 학습된 모델의 내용 확인해 보기

- $y(\text{자전거렌탈대수}) = a1(\text{temp}) + a2(\text{atemp}) + b$
- $y(\text{자전거렌탈대수}) = 8.19865874(\text{temp}) + 0.90720808(\text{atemp}) + 4.24813264$

```
In [11]: model.coef_
```

```
Out[11]: array([8.19865874, 0.90720808])
```

```
In [12]: model.intercept_
```

```
Out[12]: 4.248132645803764
```

## 1-5 학습된 모델로 예측 후, 이값으로 제출하기

```
In [13]: pred = model.predict(X_test) # 예측
sub = pd.read_csv("../data/bike/sampleSubmission.csv") # 답지 가져오기
sub['count'] = pred # 답쓰기
```


## 처음 만는 제출용 csv 파일

- index=False : csv 파일 행번호 없애기

```
In [14]: sub.to_csv("../data/bike/firstsubmission.csv", index=False)
```

## 제출하기

- 캐글 사이트 접속 후, 로그인
- 맨 상단에 Search에 Bike Sharing demand로 입력 후, 검색 되는 것 중 하나를 선택
- 들어간 사이트에서 대회로 접속 후,
  - 또는 다음 링크로 접속 : <https://www.kaggle.com/c/bike-sharing-demand>
- Late Submission 선택 후, 제출 영역에 csv 파일을 마우스 드래그하여 올려 제출
- 제출 후, 아래 'Make Submission' 을 버튼을 선택하면 제출 결과가 약간 후 보임.

 업로드가 완료된 후, 아래 버튼 선택 