# Kaggle 입문하기 - 데이터 분석 입문

## 학습 내용

- 캐글에 대해 이해하기
- 캐글 데이터 셋을 이용하여 데이터 분석을 이해한다.

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

In [1]:

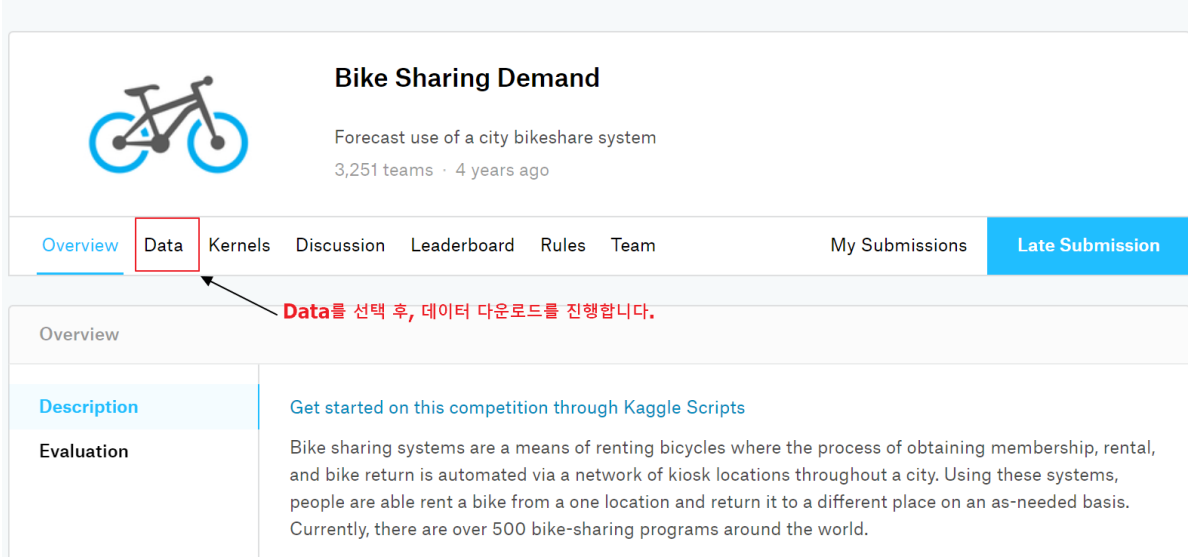
```
from IPython.display import display, Image
```

# 데이터 다운로드하기

- 가. https://www.kaggle.com/c/bike-sharing-demand (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.

In [2]:

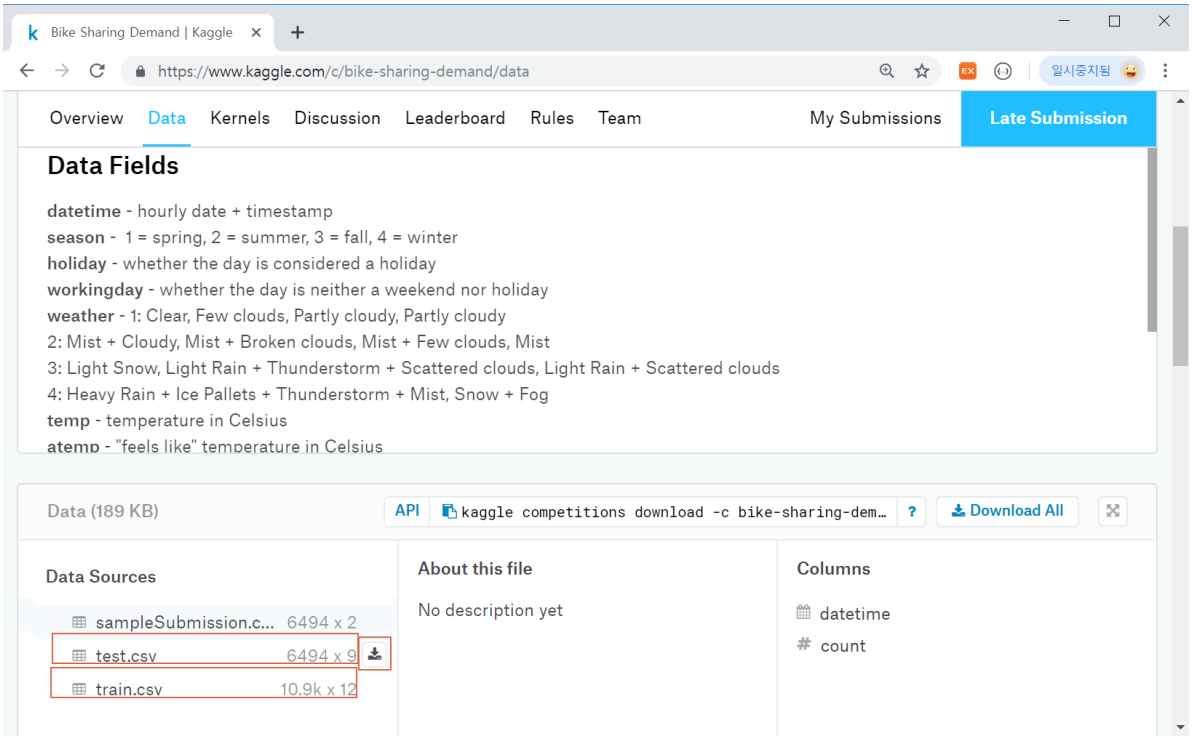
```
display(Image(filename='img/kaggle/kaggle01.png'))
```




- 'Data'를 누르면 데이터 상세 내역이 확인가능합니다.

In [3]:

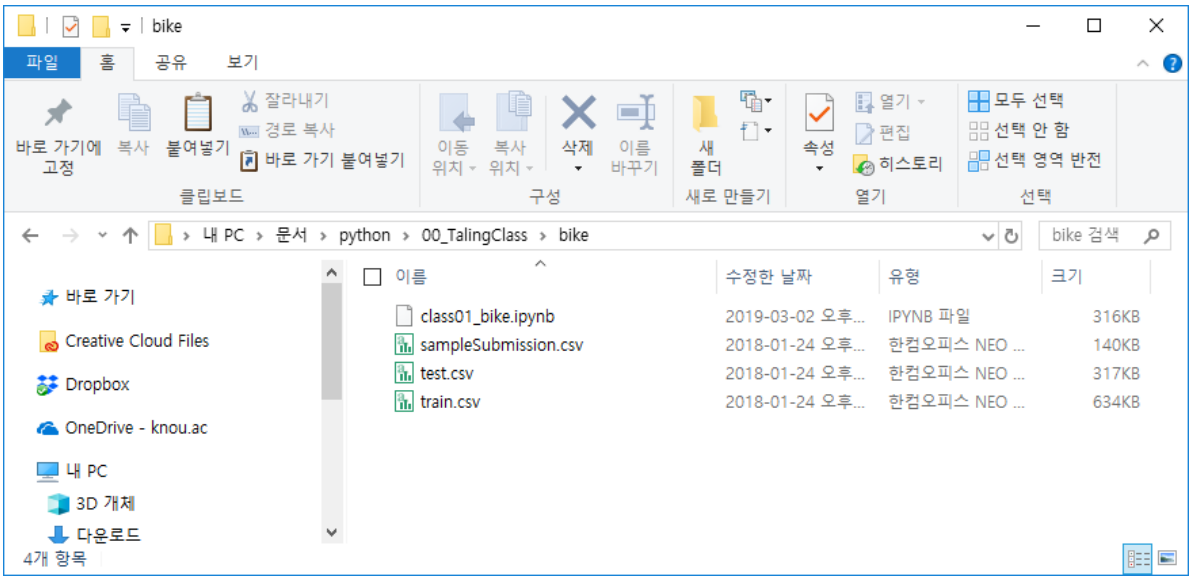
```
display(Image(filename='img/kaggle/kaggle02.png'))
```




- 'Data Sources'의 test.csv와 train.csv의 데이터 셋을 다운로드 합니다.

In [4]:

```
display(Image(filename='img/kaggle/kaggle03.png'))
```




# 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<br>2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist<br>3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds<br>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) |
| registered | number of registered user rentals initiated (가입자 사용유저) |
| count | number of total rentals (전체 렌탈 대수) |

In [5]:

```
import pandas as pd
```

## 1-1 데이터 준비하기

- 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 [6]:

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

In [7]:

```
print(train.shape)   # : 행과 열 갯수 확인
print(test.shape)
```

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

In [8]:

```
train.head()
```

Out[8]:

| | datetime | season | holiday | workingday | weather | temp | atemp | humidity | windspeed | casual |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2011-01-01 00:00:00 | 1 | 0 | 0 | 1 | 9.84 | 14.395 | 81 | 0.0 | 3 |
| **1** | 2011-01-01 01:00:00 | 1 | 0 | 0 | 1 | 9.02 | 13.635 | 80 | 0.0 | 8 |
| **2** | 2011-01-01 02:00:00 | 1 | 0 | 0 | 1 | 9.02 | 13.635 | 80 | 0.0 | 5 |
| **3** | 2011-01-01 03:00:00 | 1 | 0 | 0 | 1 | 9.84 | 14.395 | 75 | 0.0 | 3 |
| **4** | 2011-01-01 04:00:00 | 1 | 0 | 0 | 1 | 9.84 | 14.395 | 75 | 0.0 | 0 |

In [9]:

```
train.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
```

## 입력데이터 선택

In [10]:

```
f_names = ['temp', 'atemp']
X_train = train[f_names]    # 학습용 데이터의 변수 선택
X_test = test[f_names]      # 테스트 데이터의 변수 선택
```

## 출력 데이터 선택

In [11]:

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

## 1-2 모델 만들기 및 제출

## 모델 만들기 및 예측 순서

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

In [12]:

```
from sklearn.linear_model import LinearRegression
```

In [13]:

```
model = LinearRegression()
model.fit(X_train, y_train)
model.predict(X_test)   # 예측(새로운 데이터로)
```

Out[13]:

```
array([101.95625474, 104.0156171 , 104.0156171 , ..., 103.33067499,
       104.0156171 , 104.0156171 ])
```

In [14]:

```
print( model.coef_ )       # 모델(선형회귀의 계수)
print( model.intercept_)   # 모델(선형 회귀의 교차점)
```

```
[8.19865874 0.90720808]
4.248132645803707
```

In [15]:

```
## 우리가 만든 모델
## 렌탈 대수 = temp * 8.19 + atemp * 0.97 + 4.24..
```

## 학습된 모델로 예측 후, 이값으로 제출하기

In [18]:

```
sub = pd.read_csv("bike/sampleSubmission.csv")
sub.head()
```

Out[18]:

| | datetime | count |
|---|---|---|
| **0** | 2011-01-20 00:00:00 | 0 |
| **1** | 2011-01-20 01:00:00 | 0 |
| **2** | 2011-01-20 02:00:00 | 0 |
| **3** | 2011-01-20 03:00:00 | 0 |
| **4** | 2011-01-20 04:00:00 | 0 |

In [19]:

```
pred = model.predict(X_test)   # 예측
sub['count'] = pred
sub
```

Out[19]:

| | datetime | count |
|---|---|---|
| **0** | 2011-01-20 00:00:00 | 101.956255 |
| **1** | 2011-01-20 01:00:00 | 104.015617 |
| **2** | 2011-01-20 02:00:00 | 104.015617 |
| **3** | 2011-01-20 03:00:00 | 103.330675 |
| **4** | 2011-01-20 04:00:00 | 103.330675 |
| **...** | ... | ... |
| **6488** | 2012-12-31 19:00:00 | 103.330675 |
| **6489** | 2012-12-31 20:00:00 | 103.330675 |
| **6490** | 2012-12-31 21:00:00 | 103.330675 |
| **6491** | 2012-12-31 22:00:00 | 104.015617 |
| **6492** | 2012-12-31 23:00:00 | 104.015617 |

6493 rows × 2 columns

## 처음 만는 제출용 csv 파일

- index=False : csv 파일 행번오 없애기

In [20]:

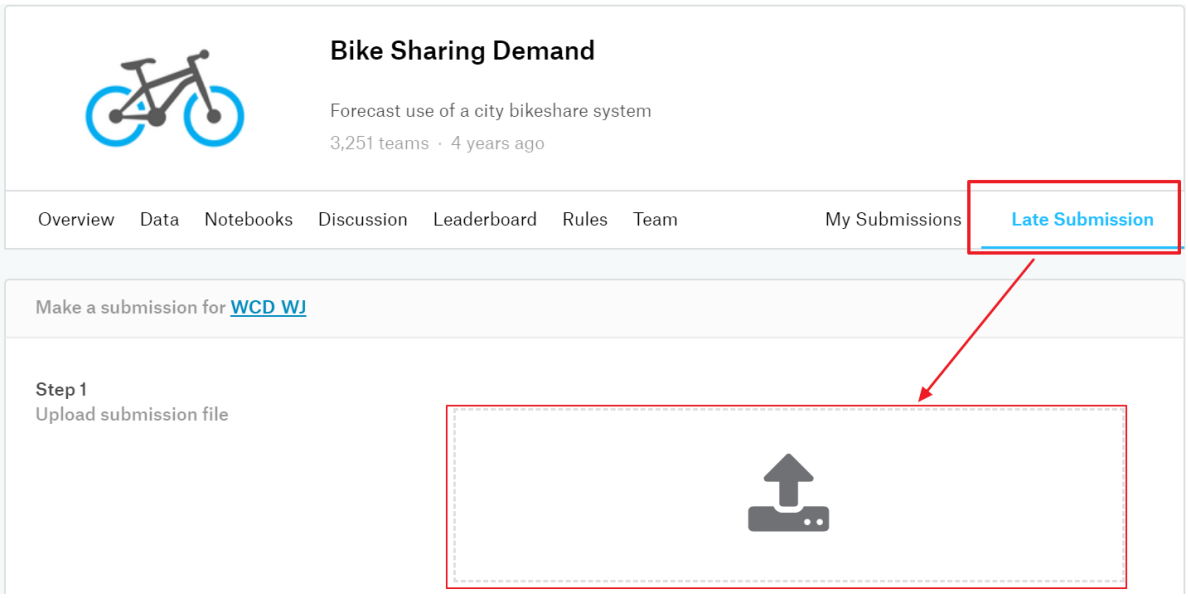
```
# 처음 만는 제출용 csv 파일, 행번호를 없애기
sub.to_csv("firstsubmission.csv", index=False)
```

## 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 (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.

In [21]:

```
display(Image(filename='img/kaggle/bike01.png'))
```




In [22]:

```
## 업로드가 완료된 후, 아래 버튼 선택
display(Image(filename='img/kaggle/bike01.png'))
```