Kaggle Bike Sharing Demand

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

- Bike_Sharing_Demand.csv (10,886 * 12)
  - datetime : Date + Timestamp
  - season : 1(봄), 2(여름), 3(가을), 4(겨울)
  - holiday : 1(토요일, 일요일을 제외한 공휴일), 0(휴일이 아닌 날)
  - workingday : 1(토요일, 일요일 및 휴일이 아닌 주중), 0(주말 및 휴일)
  - weather : 1(맑음, 약간흐림), 2(안개, 흐림), 3(가벼운 눈/비 + 천둥), 4(심한 눈/비, 천둥/번개)
  - temp : 온도(섭씨)
  - atemp : 체감온도(섭씨)
  - humidity : 습도
  - windspeed : 풍속
  - casual : 사전 등록되지 않은 사용자 대여 횟수
  - registered : 사전 등록된 사용자 대여 횟수
  - count : 총 대여 횟수 -> y

```
import warnings
warnings.filterwarnings('ignore')
```

Data Load

- 'Bike_Sharing_Demand.csv' Github에서 읽어오기

```
import pandas as pd

url = 'https://raw.githubusercontent.com/rusita-ai/pyData/master/Bike_Sharing_Demand.csv'
DF = pd.read_csv(url)

DF.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  object 
 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: float64(3), int64(8), object(1)
memory usage: 1020.7+ KB
```

```
DF.head()
```

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspee
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.
	2011-01-								

I. Data Preprocessing

1) 'String' -> 'Date'

- "year', 'month', 'day', 'hour' 추출

```
DF['datetime'] = DF.datetime.apply(pd.to_datetime)

DF['year'] = DF.datetime.apply(lambda x : x.year)
DF['month'] = DF.datetime.apply(lambda x : x.month)
DF['day'] = DF.datetime.apply(lambda x : x.day)
DF['hour'] = DF.datetime.apply(lambda x: x.hour)
```

```
DF.head()
```

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspee
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.
	2011-01-								

2) Drop Columns

- 'datetime', 'casual', 'registered'

```
drop_columns = ['datetime', 'casual', 'registered']

DF.drop(drop_columns, axis = 1,inplace = True)
```

```
DF.head()
```

	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	count	y
0	1	0	0	1	9.84	14.395	81	0.0	16	2
1	1	0	0	1	9.02	13.635	80	0.0	40	2
2	1	0	0	1	9.02	13.635	80	0.0	32	2
3	1	0	0	1	9.84	14.395	75	0.0	13	2
4	1	0	0	1	9.84	14.395	75	0.0	1	2

#

#

#

The End

#

#

#