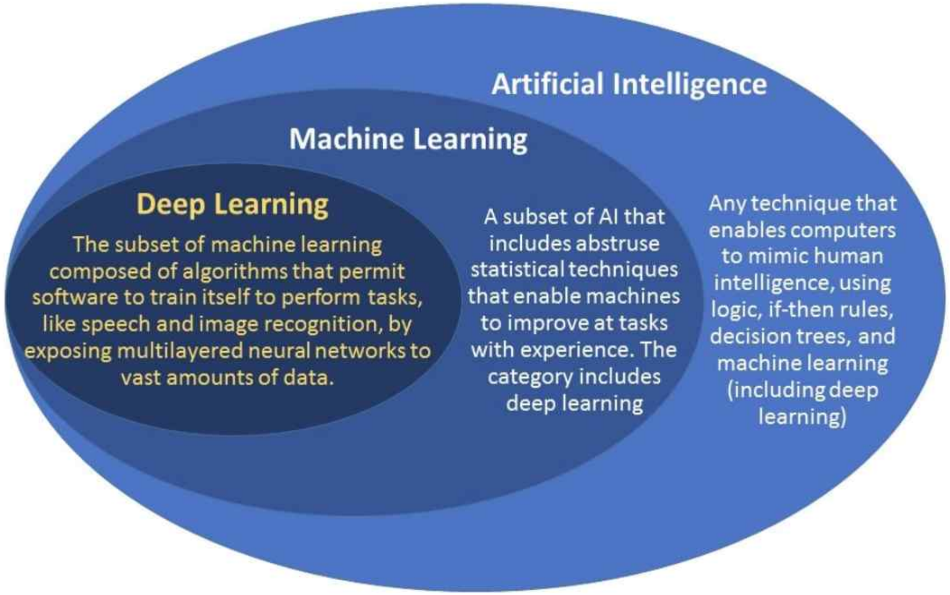
공공 자전거 수요 예측

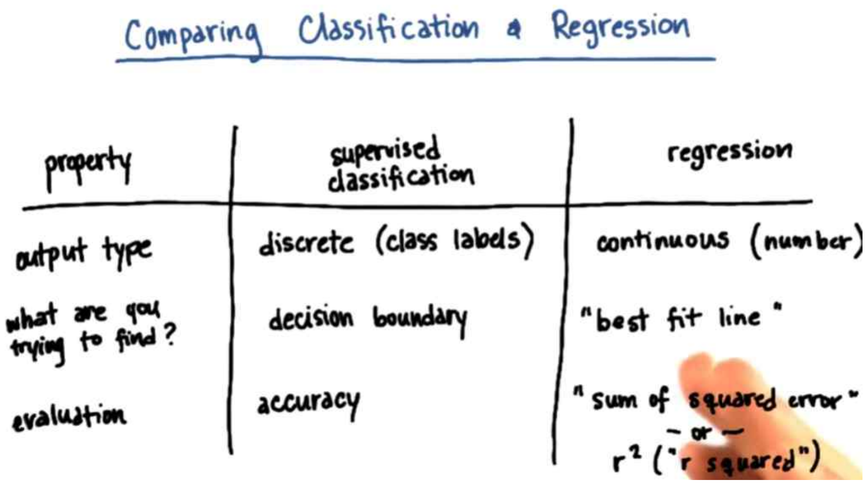
Daegu Campus of
KOREA POLYTECHNIC

Why we learn the Machine Learning?


Artificial Intelligence
Machine Learning
Deep Learning
The subset of machine learning composed of algorithms that permit software to train itself to perform tasks, like speech and image recognition, by exposing multilayered neural networks to vast amounts of data.
A subset of AI that includes abstruse statistical techniques that enable machines to improve at tasks with experience. The category includes deep learning
Any technique that enables computers to mimic human intelligence, using logic, if-then rules, decision trees, and machine learning (including deep learning)

Types of the Machine Learning

- Several types of learning algorithm
  - **Supervised Learning**
    - **Training (Learning) with labeled data**
    - **Type of Supervised Learning**
      - **Regression**
        - **Predicts final exam score based on time spent**
      - Binary Classification
        - Pass/fail based on time spent
      - Multi-label Classification
        - Grade (A, B, C, D and F) based on time spent
      - Detection, Segmentation, etc.
  - Unsupervised Learning
  - Reinforcement learning
  - Recommender systems
  - Etc.



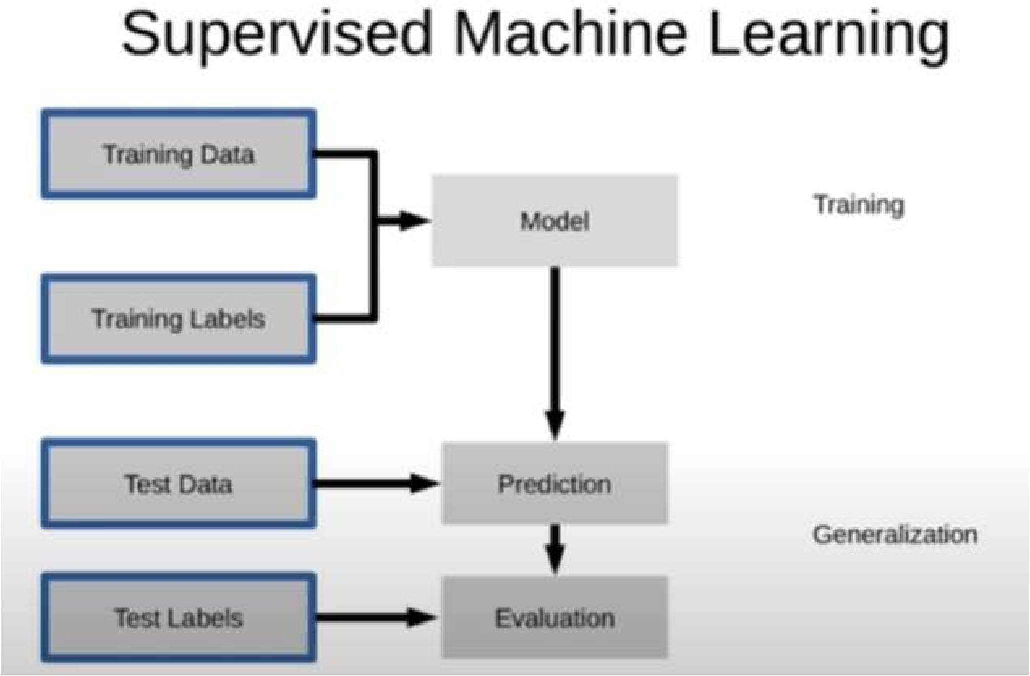
Comparing Classification & Regression

property	supervised classification	regression
output type	discrete (class labels)	continuous (number)
what are you trying to find?	decision boundary	"best fit line"
evaluation	accuracy	"sum of squared error" — or — r² ("r squared")

[출처] https://www.youtube.com/watch?v=G_0W912qmGc

Linear Regression

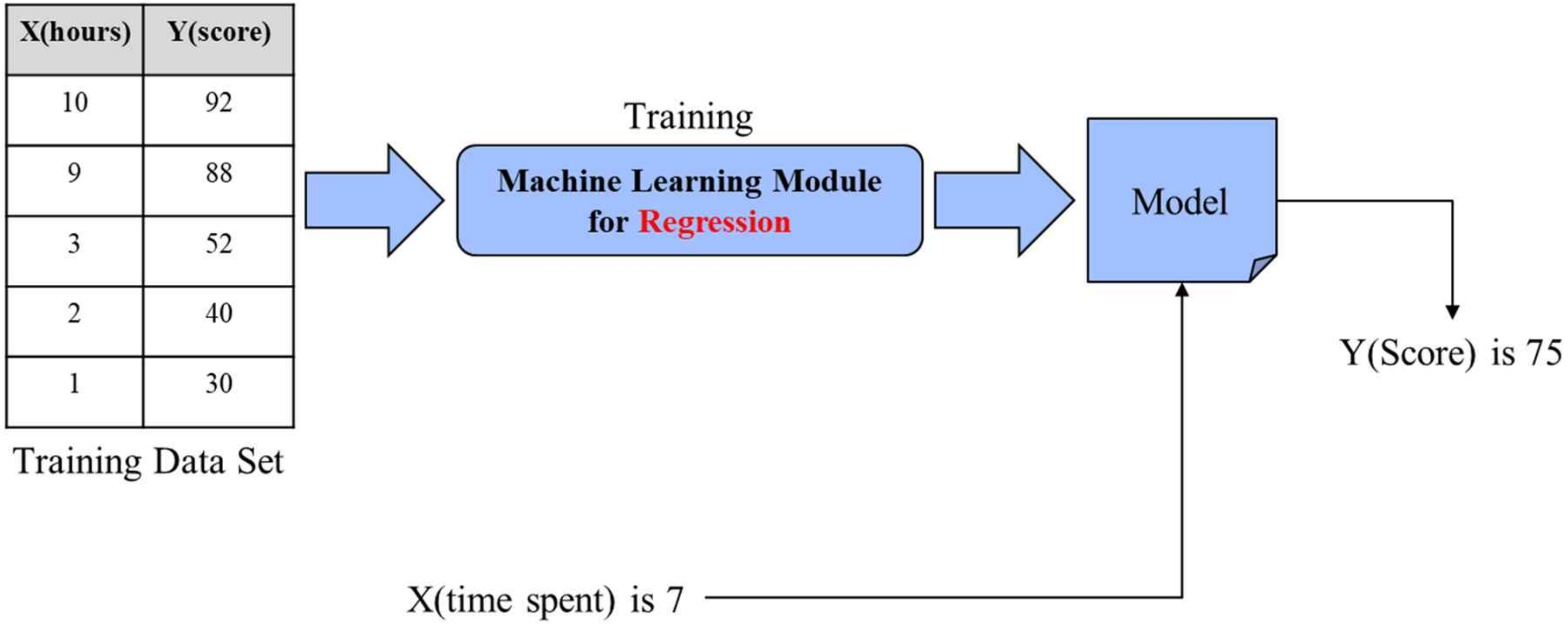
- Supervised Machine Learning



출처: https://github.com/amueller/odscon-2015

Linear Regression

- Problem: Predicts final exam score based on time spent
  - Make (obtain) Training Data Set
  - Training using training data set
  - Estimates the final exam score

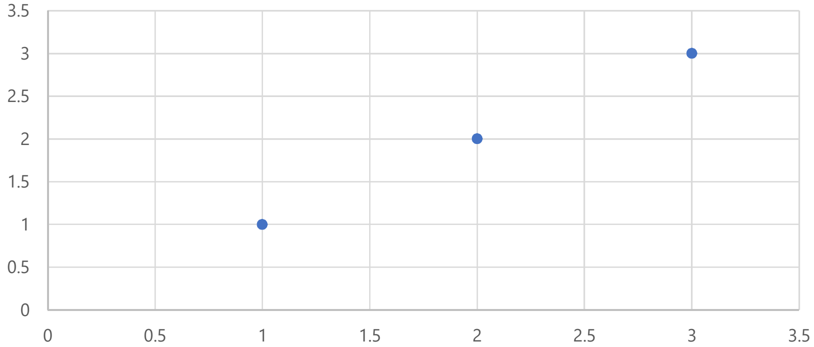


X(hours)	Y(score)
10	92
9	88
3	52
2	40
1	30

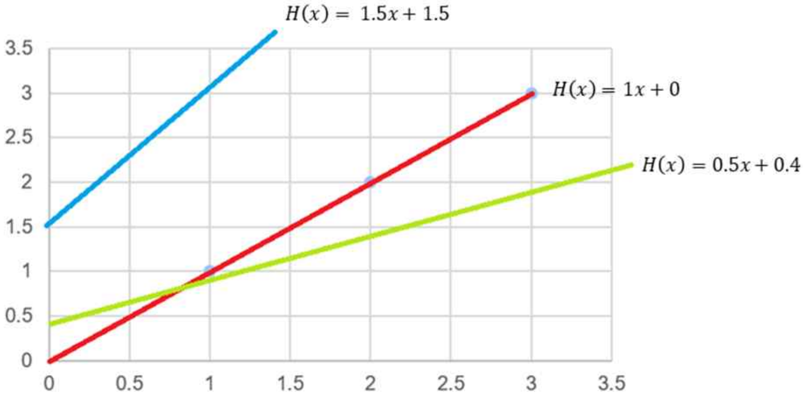
Linear Regression

- Data 단순화

X	Y
1	1
2	2
3	3



- **Hypothesis (Linear)**
  - H(x) = Wx + b

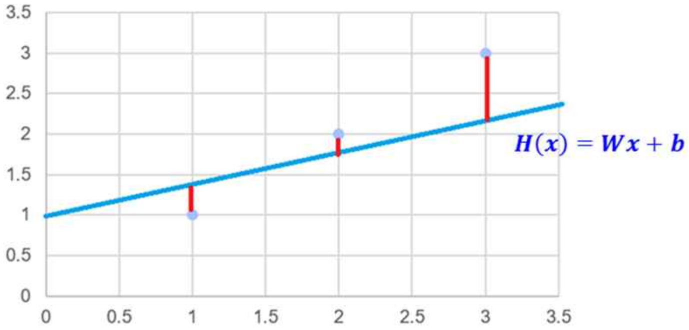


Which H(x) is better?

Linear Regression

- Cost Function (Loss Function): 실제 데이터와 가설 함수가 얼마나 차이가 나는가?
  - **How Fit the line to our training data**

$$\frac{(H(x^{(1)}) - y^{(1)})^2 + (H(x^{(2)}) - y^{(2)})^2 + (H(x^{(3)}) - y^{(3)})^2}{3}$$



$$cost = \frac{1}{m}\sum_{i=1}^{m}(H(x^{(i)}) - y^{(i)})^2$$

$$H(x) = Wx + b$$

$$\boxed{cost(W,b) = \frac{1}{m}\sum_{i=1}^{m}(H(x^{(i)}) - y^{(i)})^2}$$

⇨ **Goal: Minimize Cost**

Linear Regression

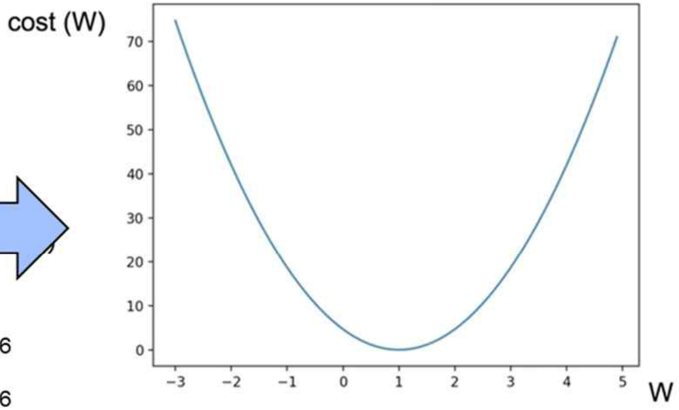
- Minimize Cost
  - Simplified Hypothesis

$$H(x) = Wx$$

$$cost(W) = \frac{1}{m}\sum_{i=1}^{m}(Wx^{(i)} - y^{(i)})^2$$

X	Y
1	1
2	2
3	3

W=0, cost = 1/3((0-1)^2+(0-2)^2+(0-3)^2)=4.666
W=1, cost = 0
W=2, cost = 1/3((2-1)^2+(4-2)^2+(6-3)^2)=4.666



e.g.) W가 5일때 시작->w값 조정->cost 계산, 0으로 수렴 할때까지 반복,

- How would you find the lowest point(minimized cost)?
  - Gradient Descent Algorithm
    - Minimize cost function
    - Be used many minimization problems
    - For a given cost function, cost(W, b), it will find W, b to minimize cost
    - It can be applied to more general function: cost(w_1, w_2, w_3, .. w_n, b)

Linear Regression

- Gradient Descent Algorithm
  - Formal Definition

$$cost(W) = \frac{1}{m}\sum_{i=1}^{m}(Wx^{(i)} - y^{(i)})^2$$

$$\Downarrow$$

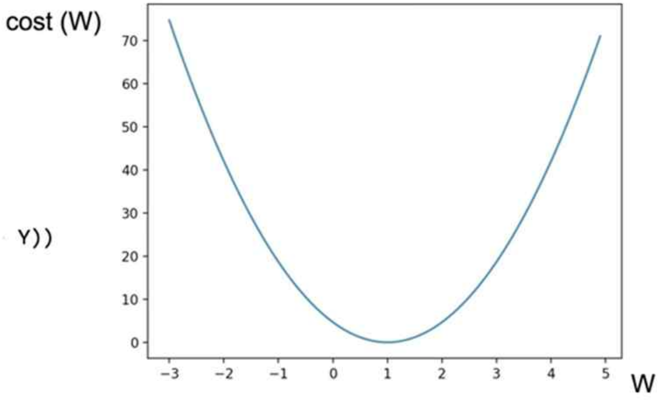
$$cost(W) = \frac{1}{2m}\sum_{i=1}^{m}(Wx^{(i)} - y^{(i)})^2$$

$$W := W - \alpha\frac{\partial}{\partial W}cost(W)$$

$$W := W - \alpha\frac{\partial}{\partial W}\frac{1}{2m}\sum_{i=1}^{m}(Wx^{(i)} - y^{(i)})^2$$

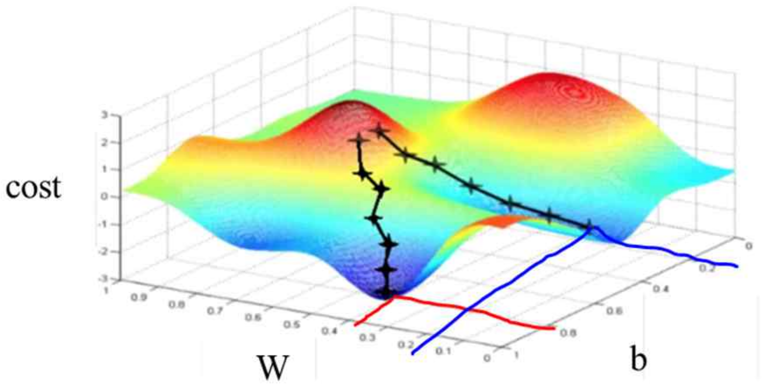
$$W := W - \alpha\frac{1}{2m}\sum_{i=1}^{m}2(Wx^{(i)} - y^{(i)})x^{(i)}$$

$$\boxed{W := W - \alpha\frac{1}{m}\sum_{i=1}^{m}(Wx^{(i)} - y^{(i)})x^{(i)}}$$

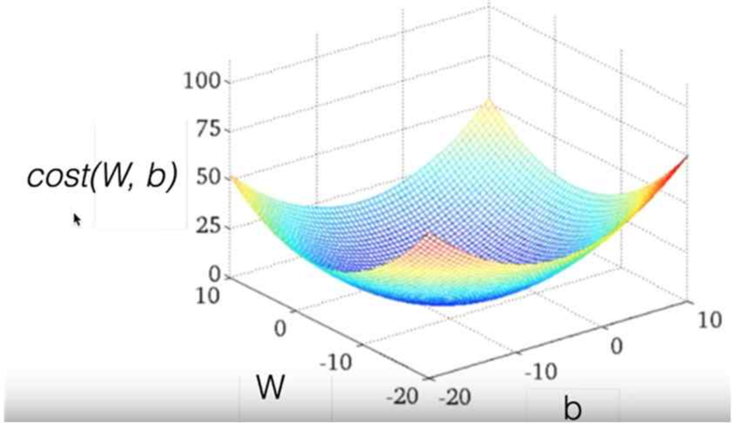


Linear Regression

- Convex Function



$$cost(W, b) = \frac{1}{m} \sum_{i=1}^{m} (H(x^{(i)}) - y^{(i)})^2$$



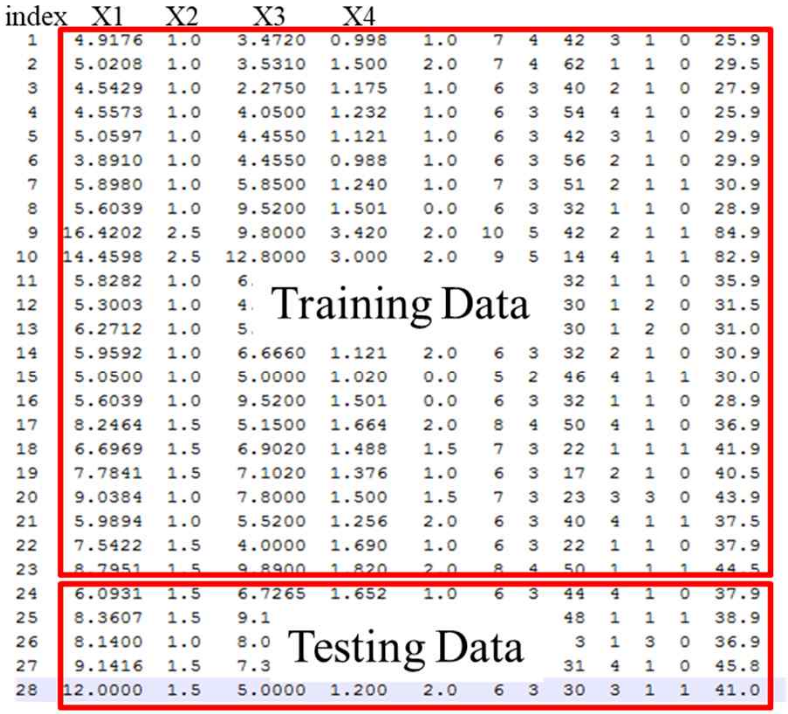
Hypothesis => Cost function => gradient Descent algorithm

출처: www.holehouse.org/mlclass

Linear Regression

- Example
  - Predict Selling price of houses

sell_house.txt


Training Data
Testing Data

index	X1	X2	X3	X4								
1	4.9176	1.0	3.4720	0.998	1.0	7	4	42	3	1	0	25.9
2	5.0208	1.0	3.5310	1.500	2.0	7	4	62	1	1	0	29.5
3	4.5429	1.0	2.2750	1.175	1.0	6	3	40	2	1	0	27.9
4	4.5573	1.0	4.0500	1.232	1.0	6	3	54	4	1	0	25.9
5	5.0597	1.0	4.4550	1.121	1.0	6	3	42	3	1	0	29.9
6	3.8910	1.0	4.4550	0.988	1.0	6	3	56	2	1	0	29.9
7	5.8980	1.0	5.8500	1.240	1.0	7	3	51	2	1	1	30.9
8	5.6039	1.0	9.5200	1.501	0.0	6	3	32	1	1	0	28.9
9	16.4202	2.5	9.8000	3.420	2.0	10	5	42	2	1	1	84.9
10	14.4598	2.5	12.8000	3.000	2.0	9	5	14	4	1	1	82.9
11	5.8282	1.0	6.					32	1	1	0	35.9
12	5.3003	1.0	4.					30	1	2	0	31.5
13	6.2712	1.0	5.					30	1	2	0	31.0
14	5.9592	1.0	6.6660	1.121	2.0	6	3	32	2	1	0	30.9
15	5.0500	1.0	5.0000	1.020	0.0	5	2	46	4	1	1	30.0
16	5.6039	1.0	9.5200	1.501	0.0	6	3	32	1	1	0	28.9
17	8.2464	1.5	5.1500	1.664	2.0	8	4	50	4	1	0	36.9
18	6.6969	1.5	6.9020	1.488	1.5	7	3	22	1	1	1	41.9
19	7.7841	1.5	7.1020	1.376	1.0	6	3	17	2	1	0	40.5
20	9.0384	1.0	7.8000	1.500	1.5	7	3	23	3	3	0	43.9
21	5.9894	1.0	5.5200	1.256	2.0	6	3	40	4	1	1	37.5
22	7.5422	1.5	4.0000	1.690	1.0	6	3	22	1	1	0	37.9
23	8.7951	1.5	9.8900	1.820	2.0	8	4	50	1	1	1	44.5
24	6.0931	1.5	6.7265	1.652	1.0	6	3	44	4	1	0	37.9
25	8.3607	1.5	9.1					48	1	1	1	38.9
26	8.1400	1.0	8.0					3	1	3	0	36.9
27	9.1416	1.5	7.3					31	4	1	0	45.8
28	12.0000	1.5	5.0000	1.200	2.0	6	3	30	3	1	1	41.0

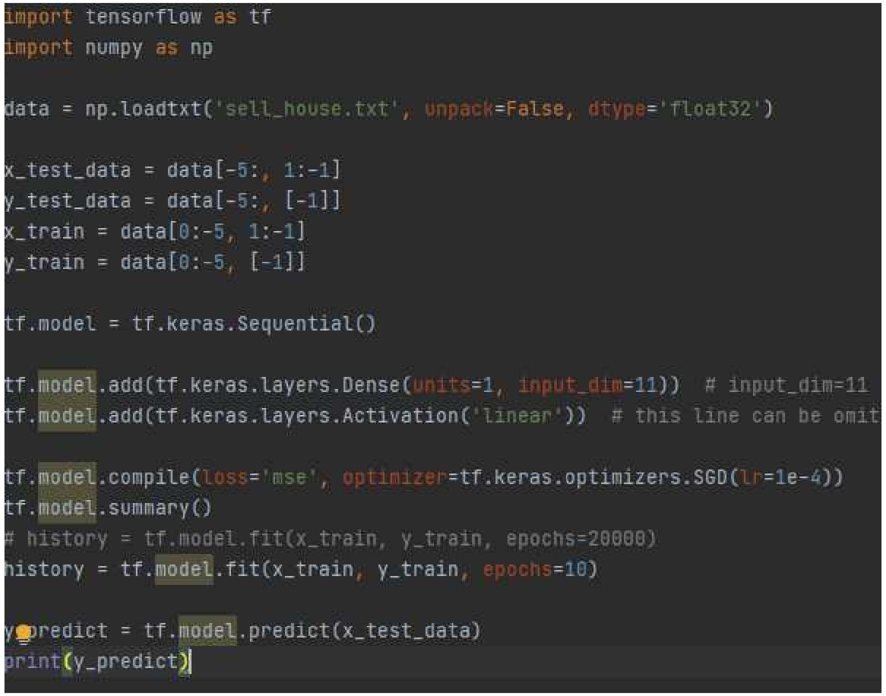
(http://people.sc.fsu.edu/~jburkardt/datasets/regression/x26.txt)

I, the index;
A1, the local selling prices, in hundreds of dollars;
A2, the number of bathrooms;
A3, the area of the site in thousands of square feet;
A4, the size of the living space in thousands of square feet;
A5, the number of garages;
A6, the number of rooms;
A7, the number of bedrooms;
A8, the age in years;
A9, 1 = brick, 2 = brick/wood, 3 = aluminum/wood, 4 = wood.
A10, 1 = two story, 2 = split level, 3 = ranch
A11, number of fire places.
B, the selling price.

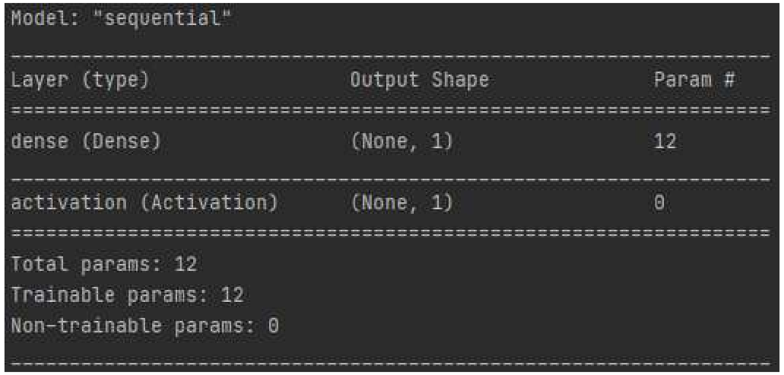
I, 색인
A1, 수백 달러의 현지 판매 가격
A2, 욕실 수
A3, 수천 평방 피트의 부지
A4, 수천 평방 피트의 생활 공간의 크기
A5, 차고 수
A6, 객실 수
A7, 침실 수
A8, 건물 년식;
A9, 1 = 벽돌, 2 = 벽돌 / 목재, 3 = 알루미늄 / 목재, 4 = 목재.
A10, 1 = 2층, 2 = 스플릿 레벨, 3 = 목장
A11, 화재 대피 장소.
B, 판매 가격

Linear Regression

- Example
  - Predict Selling price of houses
  - pip install tensorflow



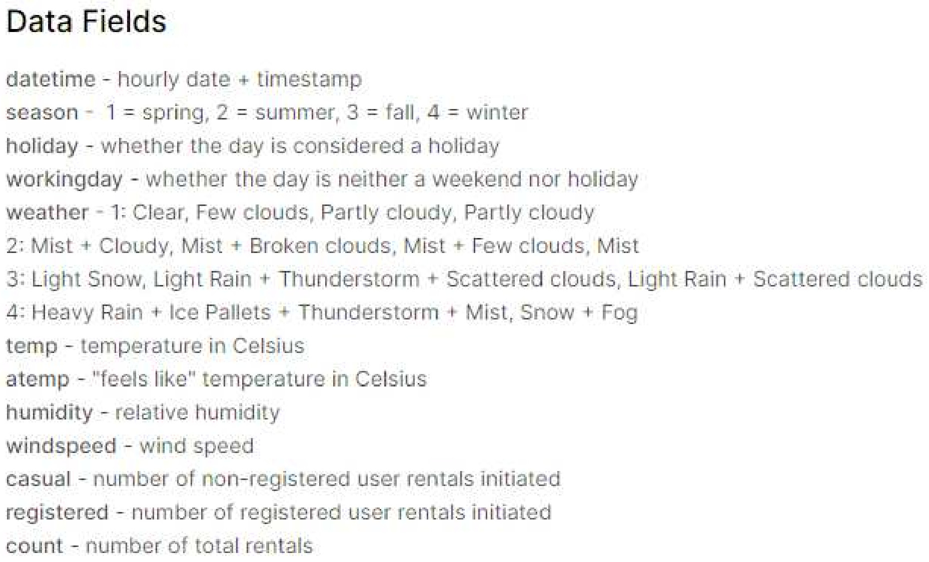
```
import tensorflow as tf
import numpy as np

data = np.loadtxt('sell_house.txt', unpack=False, dtype='float32')

x_test_data = data[-5:, 1:-1]
y_test_data = data[-5:, [-1]]
x_train = data[0:-5, 1:-1]
y_train = data[0:-5, [-1]]

tf.model = tf.keras.Sequential()

tf.model.add(tf.keras.layers.Dense(units=1, input_dim=11))  # input_dim=11
tf.model.add(tf.keras.layers.Activation('linear'))  # this line can be omit

tf.model.compile(loss='mse', optimizer=tf.keras.optimizers.SGD(lr=1e-4))
tf.model.summary()
# history = tf.model.fit(x_train, y_train, epochs=20000)
history = tf.model.fit(x_train, y_train, epochs=10)

y_predict = tf.model.predict(x_test_data)
print(y_predict)
```



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 1)                 12
_________________________________________________________________
activation (Activation)      (None, 1)                 0
=================================================================
Total params: 12
Trainable params: 12
Non-trainable params: 0
_________________________________________________________________
```

공공 자전거 수요 예측

- https://www.kaggle.com/c/bike-sharing-demand/overview
  - Training Data의 **"특징"**을 바탕으로 Test Dataset의 Count(대여량)을 예측하는 경진대회



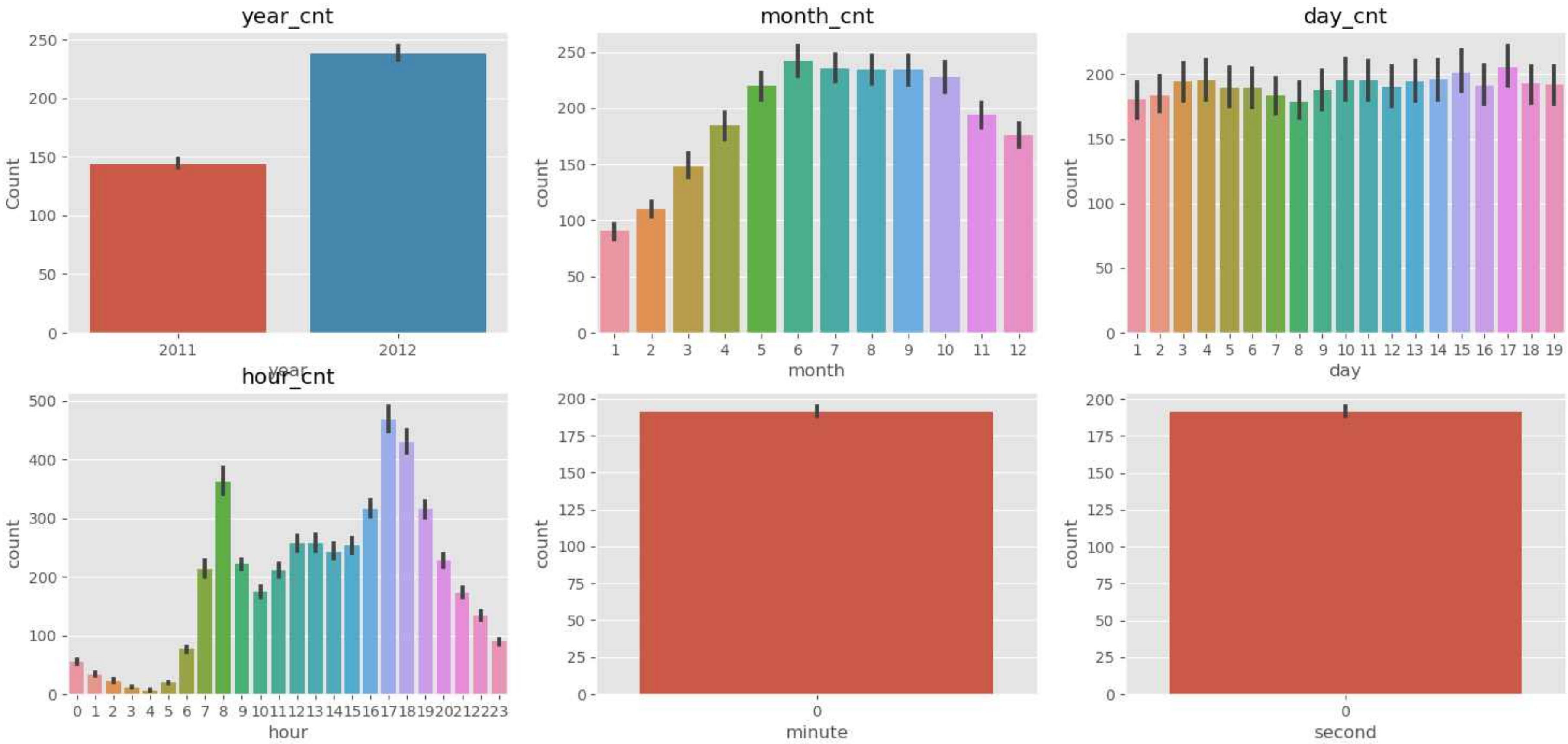
Data Fields

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
atemp - "feels like" temperature in Celsius
humidity - relative humidity
windspeed - wind speed
casual - number of non-registered user rentals initiated
registered - number of registered user rentals initiated
count - number of total rentals

- 데이터 분석
- scikit learn library ➔ Feature Engineering
- 자전거 대여량 예측
  - 분류 vs. 회귀 ➔ Random Forest-Regressor, etc.

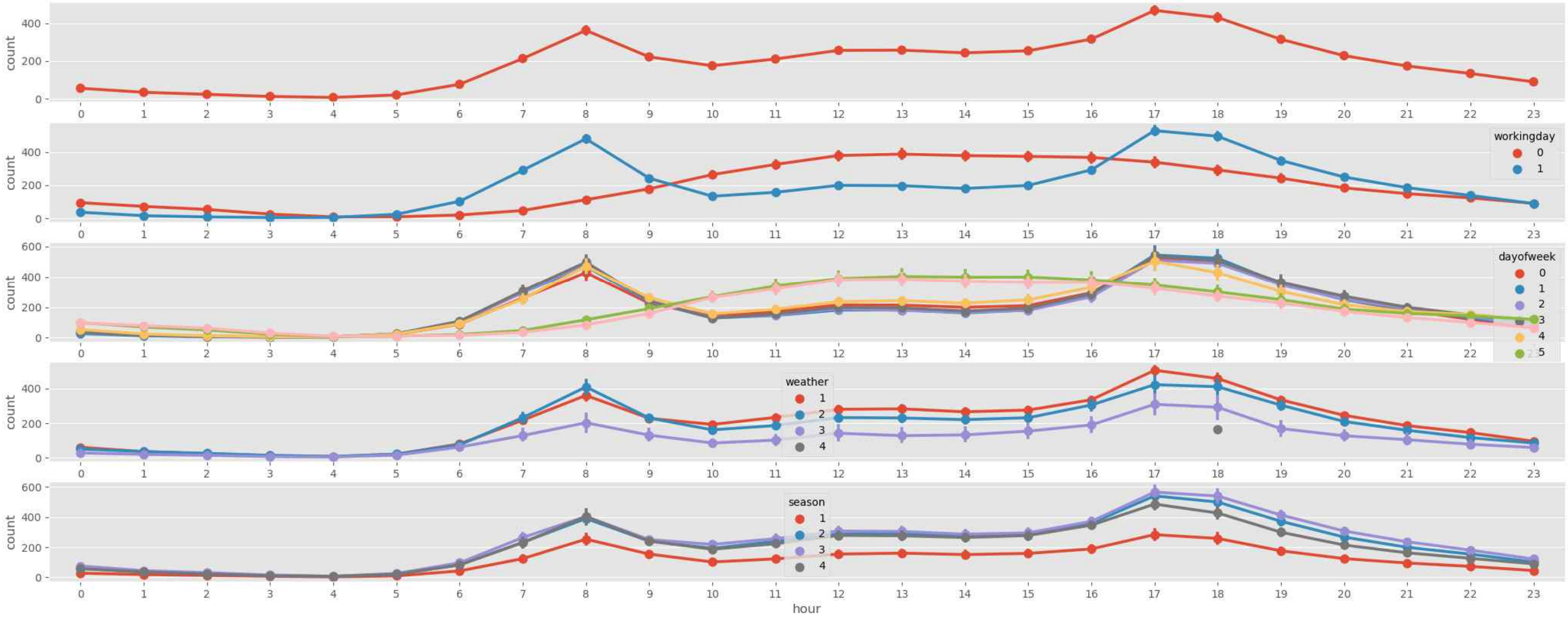
공공 자전거 수요 예측

- 데이터 분석 실습



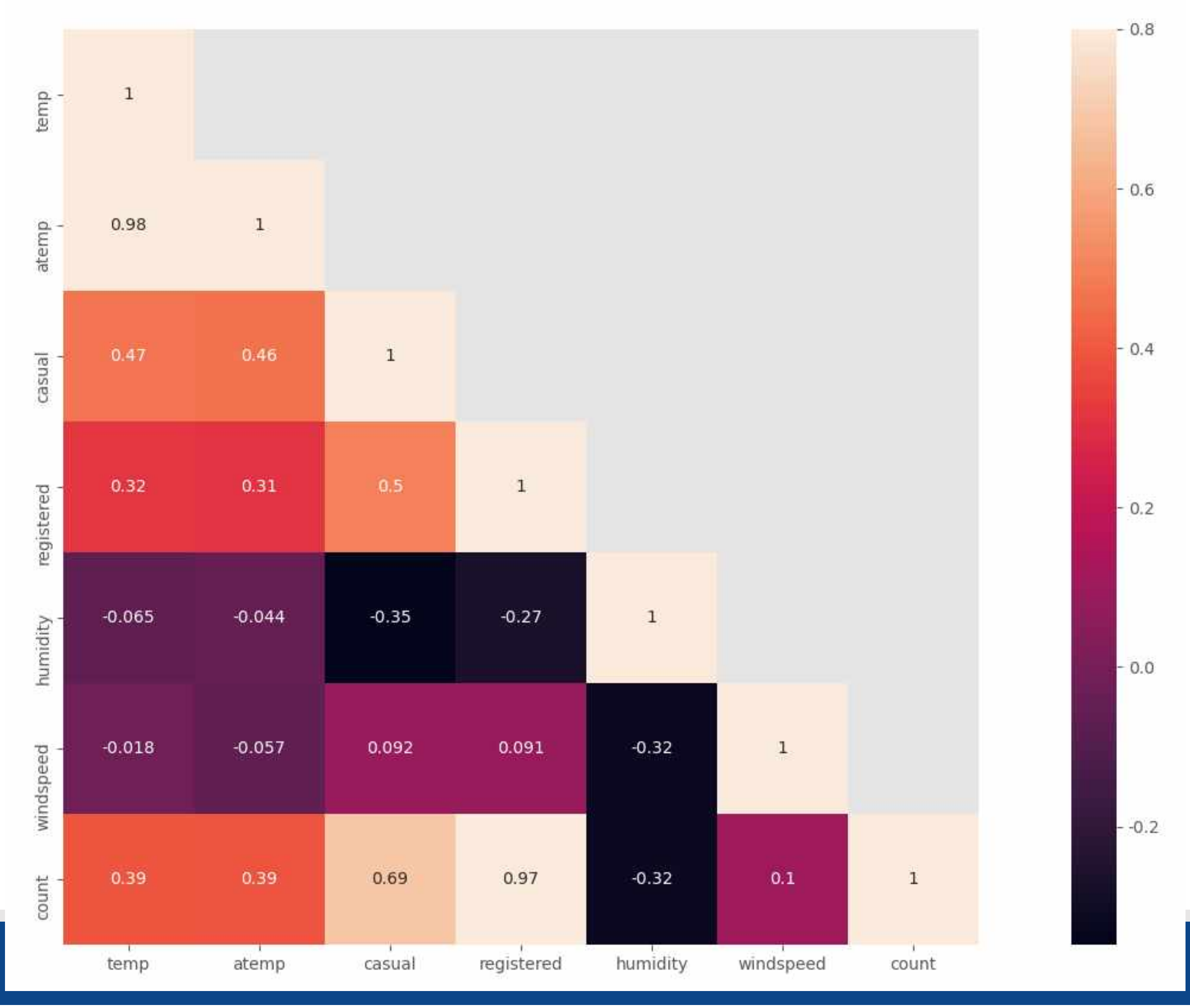
공공 자전거 수요 예측

- 데이터 분석 실습



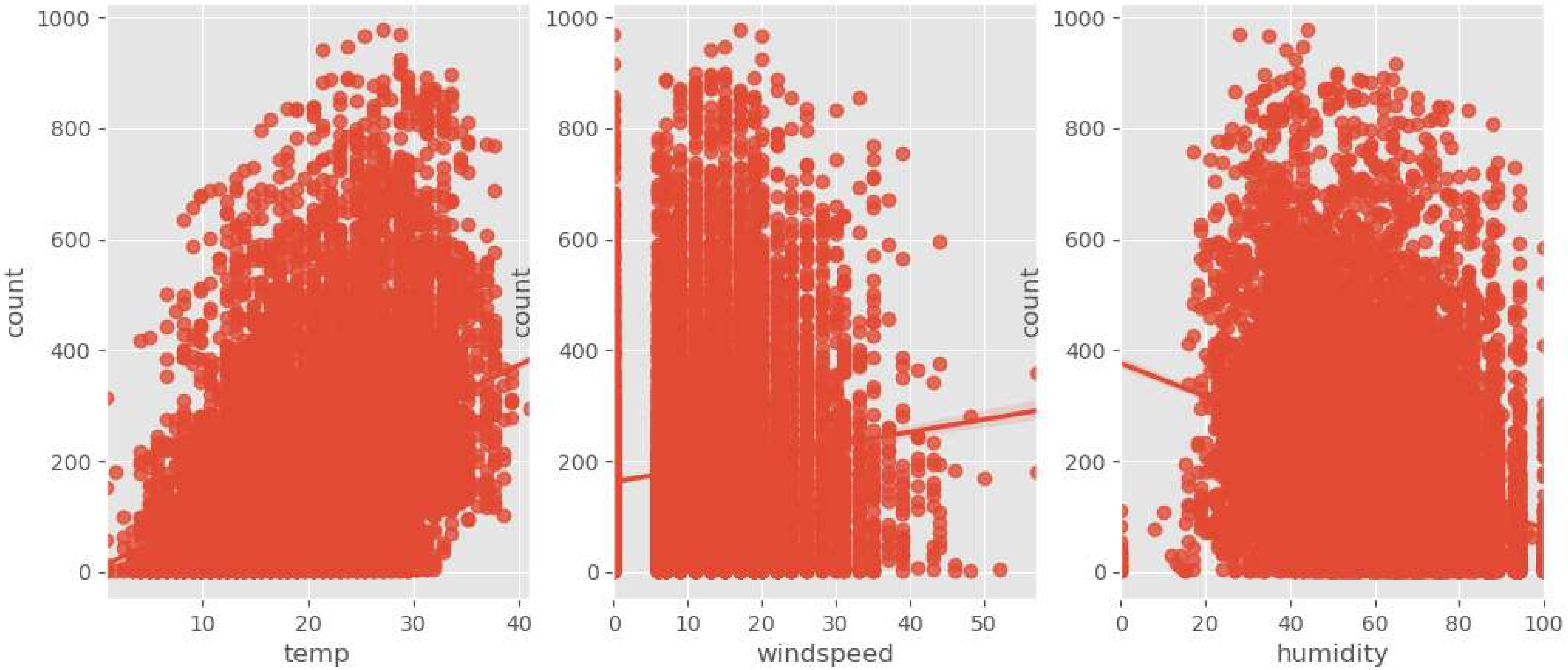
공공 자전거 수요 예측

- 데이터 분석 실습



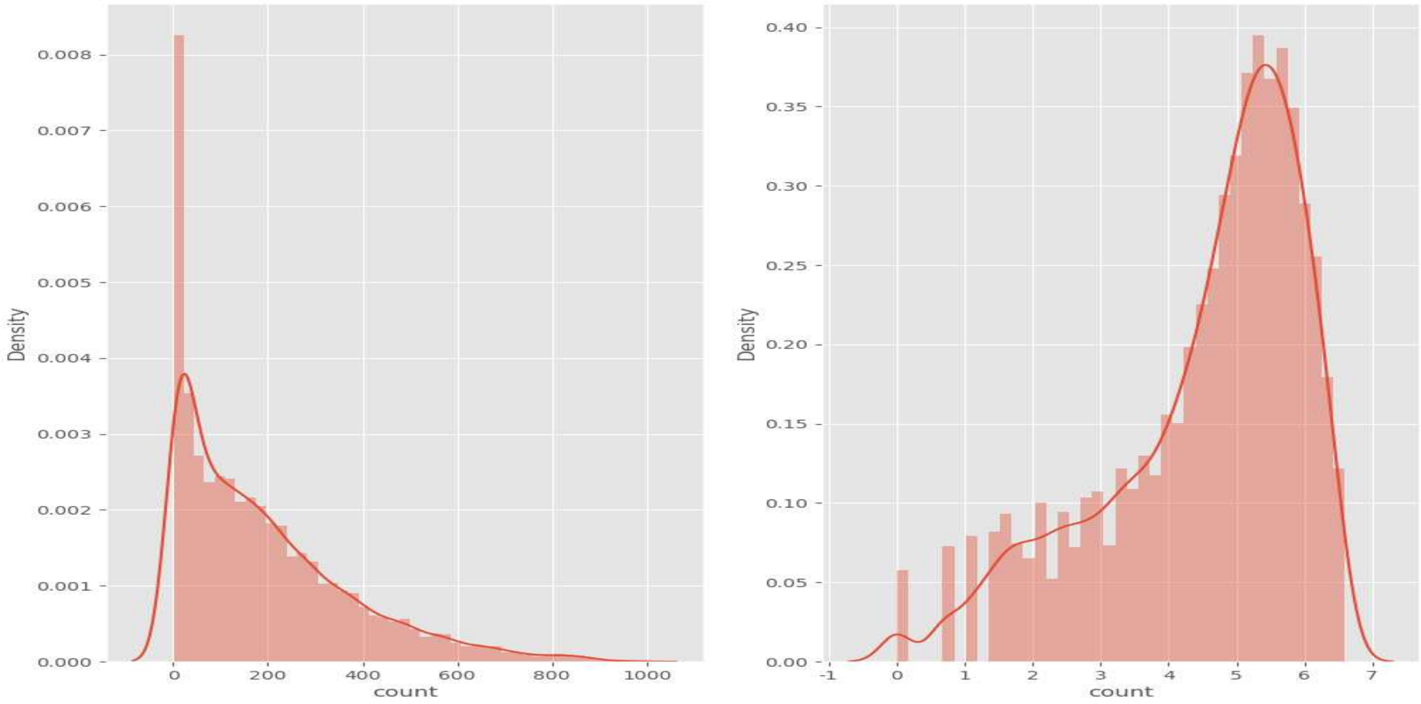
공공 자전거 수요 예측

- 데이터 분석 실습



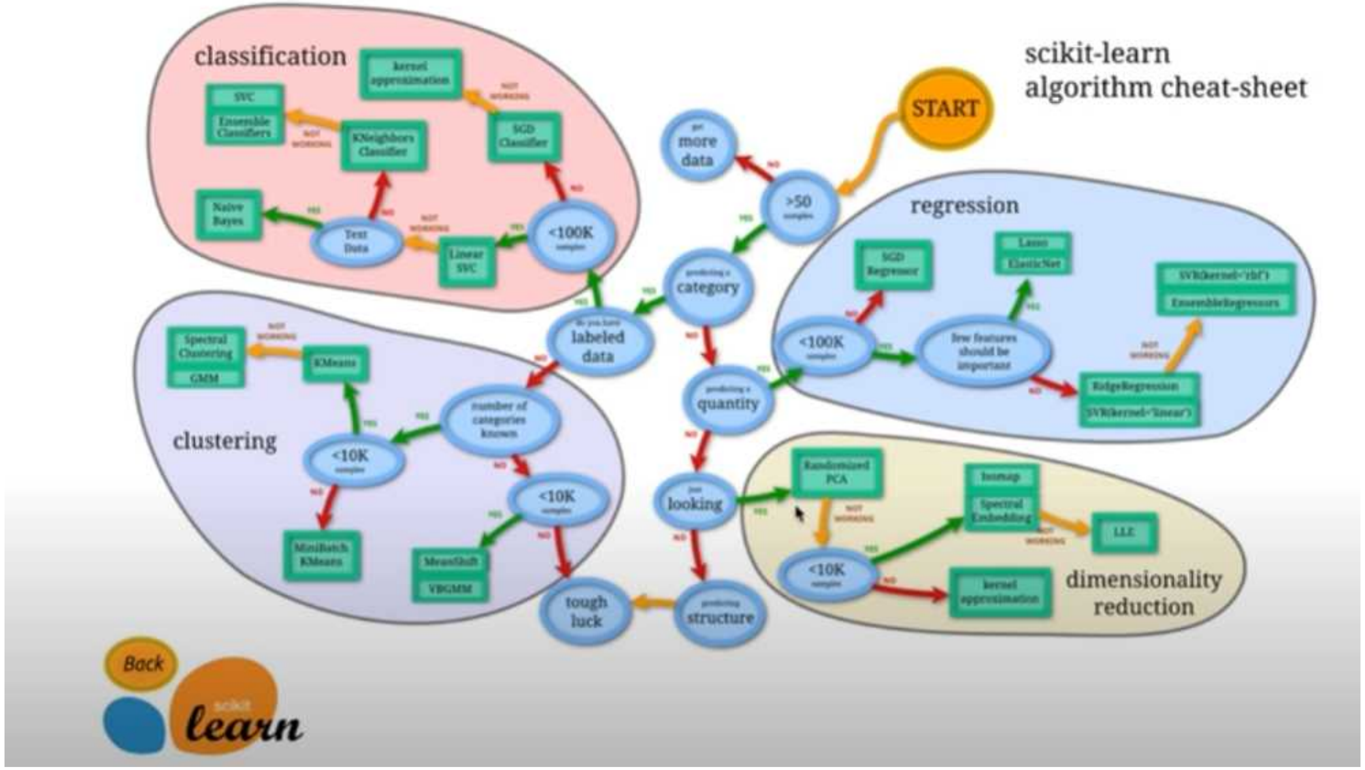
공공 자전거 수요 예측

- 데이터 분석 실습



공공 자전거 수요 예측

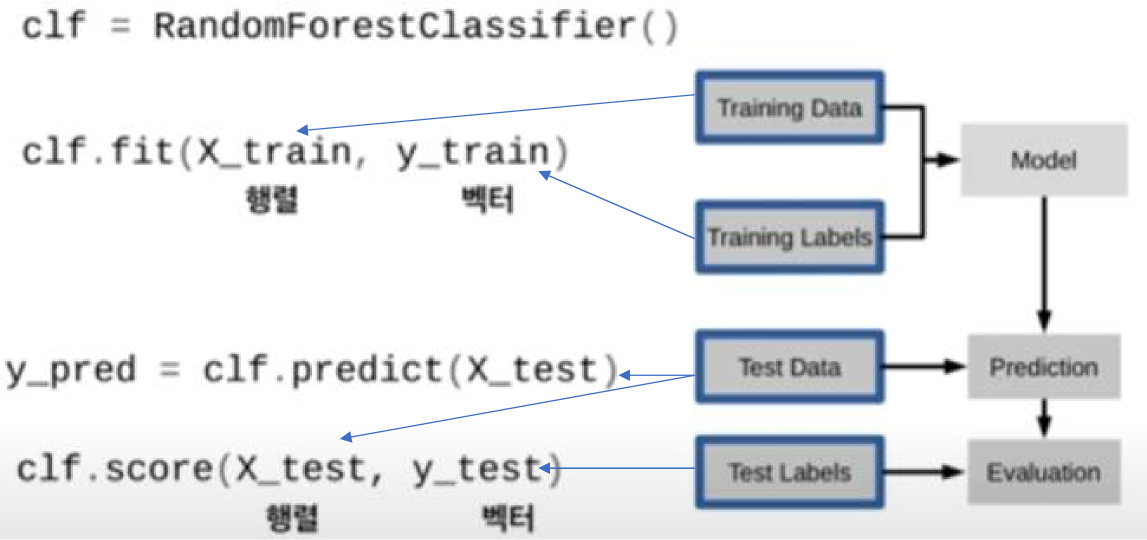
- Scikit Learn Lib.
  - ML Library



Google: scikit learn 검색

공공 자전거 수요 예측

- Scikit Learn Lib.
  - ML Library



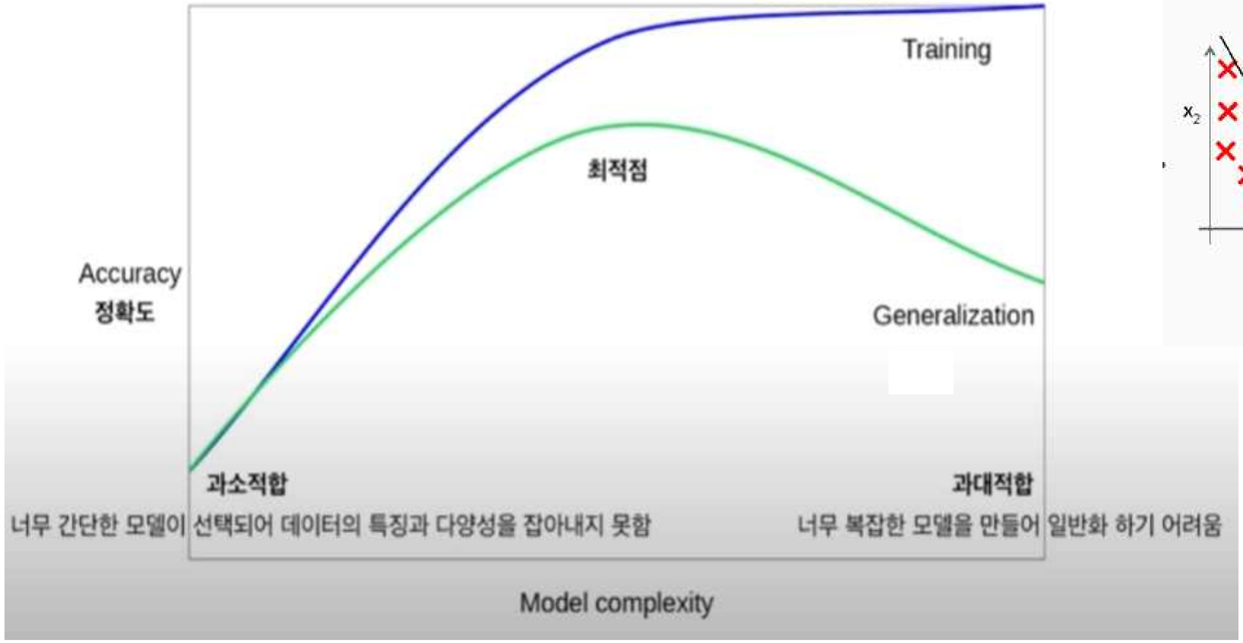
```
clf = RandomForestClassifier()

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

clf.score(X_test, y_test)
```

예측 결과 확인

모델 성능 평가

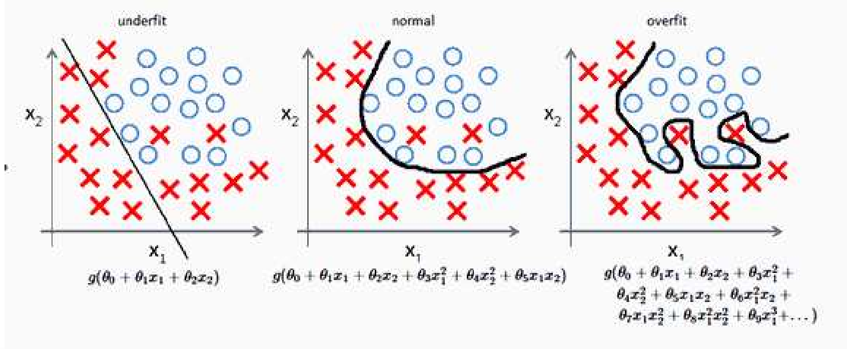
공공 자전거 수요 예측

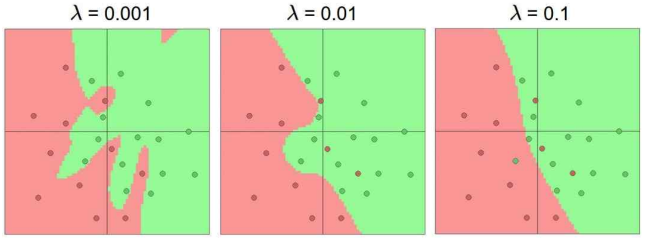
- Scikit Learn Lib.
  - Evaluation Model
  - Overfitting and Underfitting



Overfitting 해결 방안:

- More Training Data
- **Reduce the number of features (Feature Engineering 필요)**
- Regularization


underfit
normal
overfit
x_2
x_1
$g(\theta_0+\theta_1 x_1+\theta_2 x_2)$
$g(\theta_0+\theta_1 x_1+\theta_2 x_2+\theta_3 x_1^2+\theta_4 x_2^2+\theta_5 x_1 x_2)$
$g(\theta_0+\theta_1 x_1+\theta_2 x_2+\theta_3 x_1^2+\theta_4 x_2^2+\theta_5 x_1 x_2+\theta_6 x_1^2 x_2+\theta_7 x_1 x_2^2+\theta_8 x_1^2 x_2^2+\theta_9 x_1^3+\ldots)$

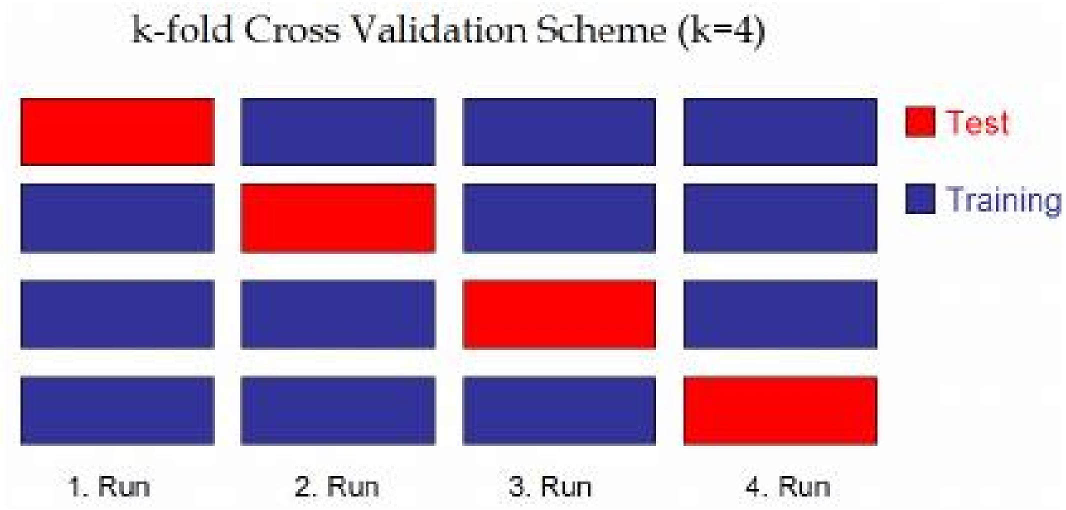


Regularization

출처: http://mlwiki.org/index.php/Overfitting

공공 자전거 수요 예측

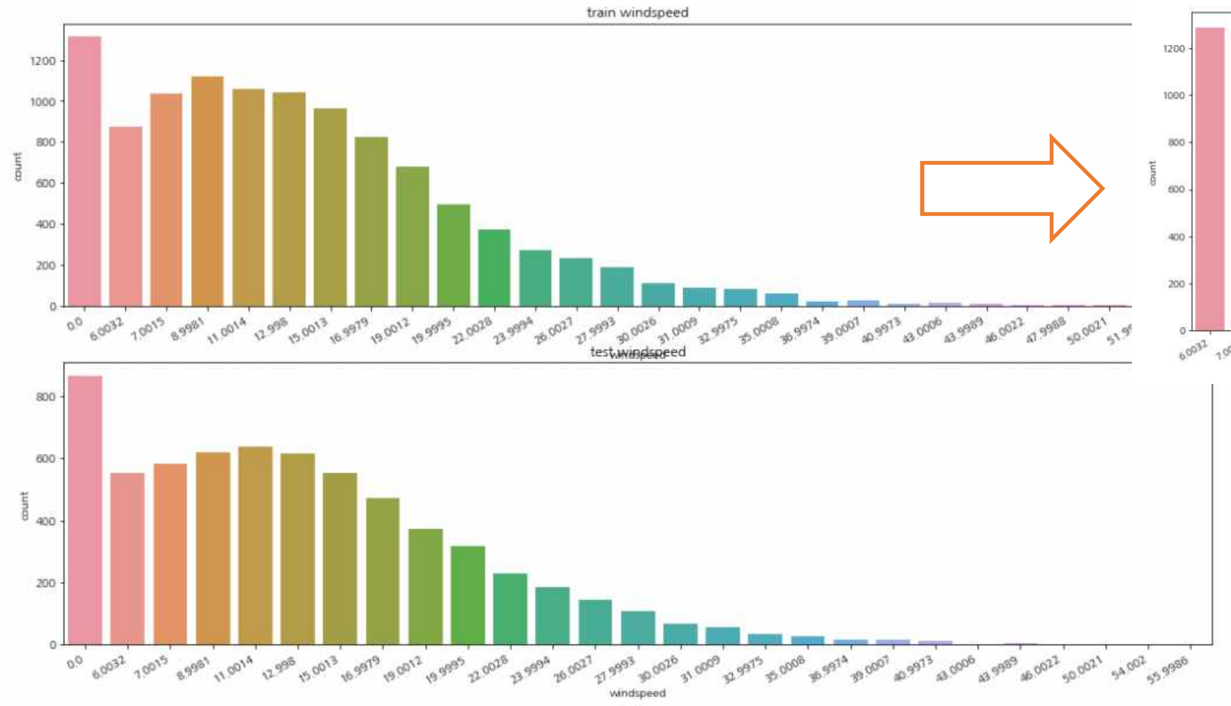
- Scikit Learn Lib.
  - Cross Validation (교차 검증)
    - 일반화 성능을 측정하기 위해 데이터를 여러 번 반복해서 나누고 여러 모델을 학습
    - 최종 정확도: 각각 정확도를 측정하여 평균 값 도출

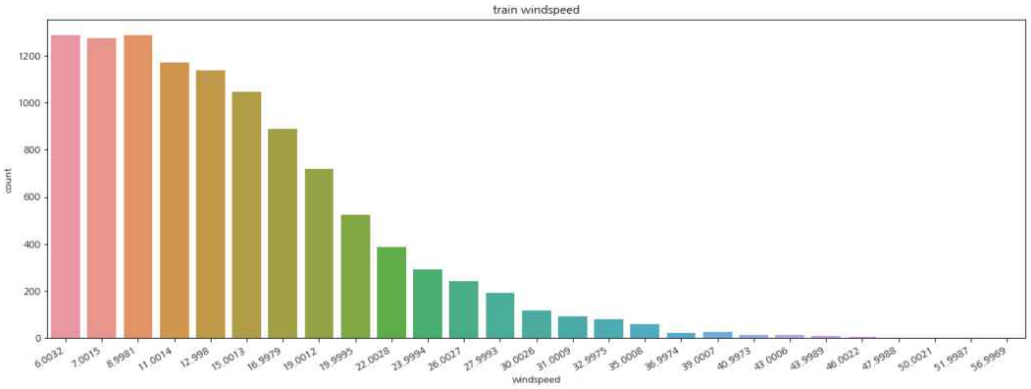


이미지 출처 : https://www.researchgate.net/figure/228403467_fig2_Figure-4-k-fold-cross-validation-scheme-example

공공 자전거 수요 예측

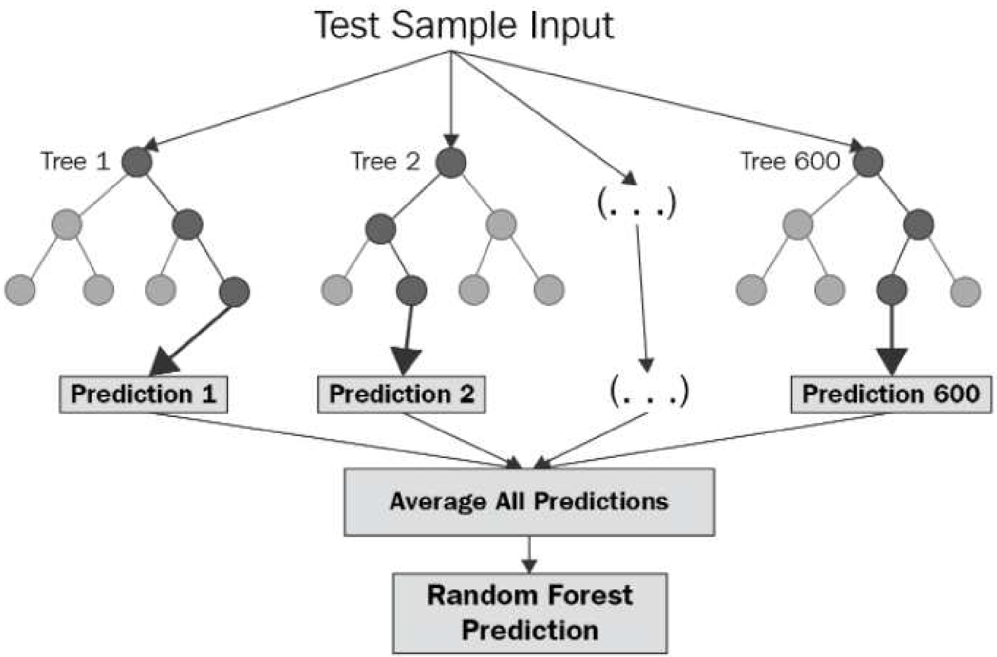
- Feature Engineering
  - 평균과 너무 떨어진 값 사용 X
  - 풍속이 0인 데이터를 적절한 데이터로 변환 (Randomforest 분류기를 이용하여 풍속 값 예측)





공공 자전거 수요 예측

- Feature Engineering
  - Random Forest: 분류, 회귀 분석 등에 사용되는 앙상블 학습 방법의 일종으로, 훈련 과정에서 구성한 다수의 결정 트리로부터 부류(분류) 또는 평균 예측치(회귀 분석)를 출력 (WIKI)



그림출처: https://post.naver.com/viewer/postView.nhn?volumeNo=28037302&memberNo=18071586

공공 자전거 수요 예측

- Feature Selection
  - 신호와 잡음을 구분
  - Feature가 많다고 무조건 좋은 성능이 나오는 것 아님 (Overfitting)
  - Feature를 추가 또는 변경해 가면서 성능이 좋지 않은 Feature는 제거

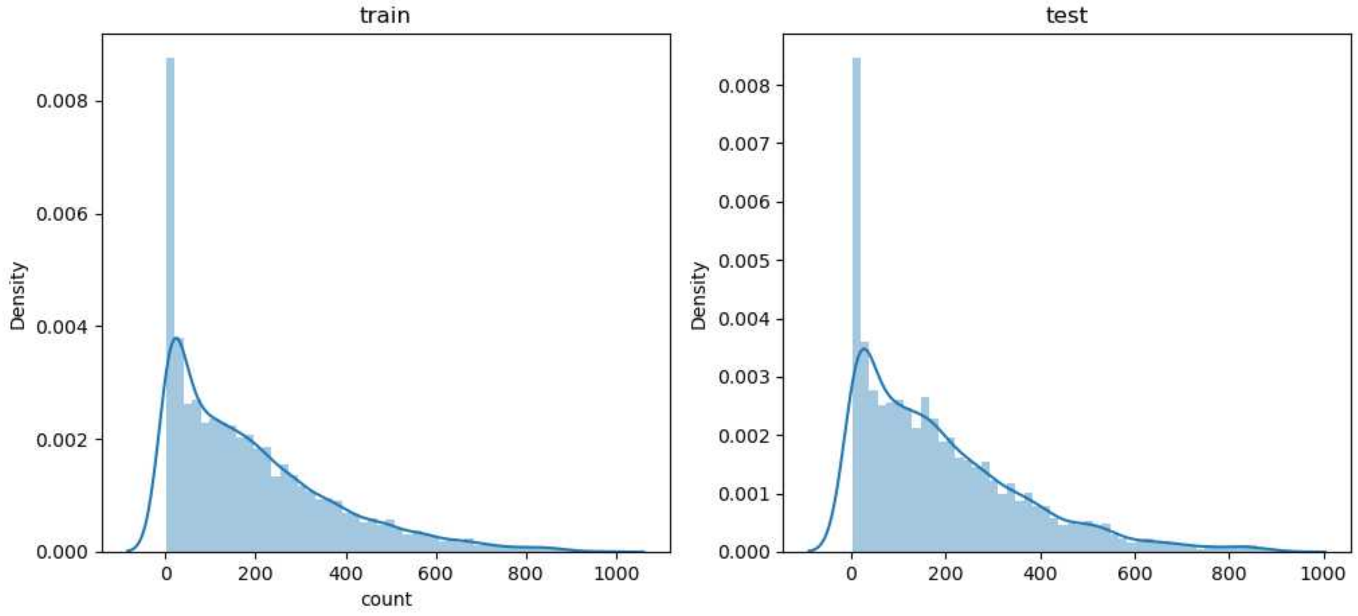
  - 연속형 Feature와 범주형 Feature 구분
    - 연속형 feature = ["temp","humidity","windspeed","atemp"]
    - 범주형 feature의 type을 category로 변경: categorical_feature_names = ["season","holiday","workingday","weather", "dayofweek","month","year","hour"]

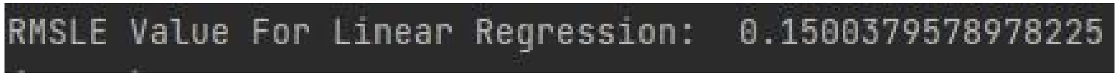
- Kaggle 공공자전거 수요 예측 평가
  - RMSLE (Root Mean Squared Logarithmic Error)

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1)-\log(a_i+1))^2}$$

공공 자전거 수요 예측

- Training & Test
  - 학습: RandomForestRegressor (Count와 같이 연속된 값은 Regressor로 학습)
  - Parameter n_estimators: decision tree의 갯수





```
RMSLE Value For Linear Regression:  0.1500379578978225
```

Thank you

Q&A

www.kopo.ac.kr
jsshin7@kopo.ac.kr