

Week2 - 예측모델링 위한 기본지식 습득

프리 온보딩 코스

에이아이스쿨(AISchool)

팀프로젝트 Week1 - 원티드 국민연금 DB를 이용해서 유니콘 기업 발굴하기

- 원티드는 크레딧잡 서비스를 운영하고 있습니다. 이 데이터는 국민연금 가입 사업장 내역이라는 공개 데이터를 기반으로 합니다. 국민연금 가입을 했다면 기록에 남을 수 밖에 없는 모든 기업 데이터를 통해 여러 인사이트를 도출 할 수 있습니다.
- 미리 진행된 데이터 전처리를 통해 기업을 식별할 수는 있는 테이블을 제공해 드립니다. 이 데이터를 조인하면 회사의 국민연금 고지금액, 매출액, 인원수를 월별로 볼 수 있습니다. 이를 통해 2015년 부터 2019년 사이의 추이를 그려보시고 **유니콘 기업으로 보이는 기업들을 나름의 방식으로 찾아주세요.**
- 이 문제에 대한 정답은 없습니다. 따라서 솔루션을 제공해 드릴 수는 없는 점 양해바랍니다.

EDA Practice #1 원티드DB 국민연금 사업장 데이터

- 원티드에서 실제로 사용 중인 데이터로, 여러 회사들과 회사들이 납부한 국민연금 보험료 정보 데이터

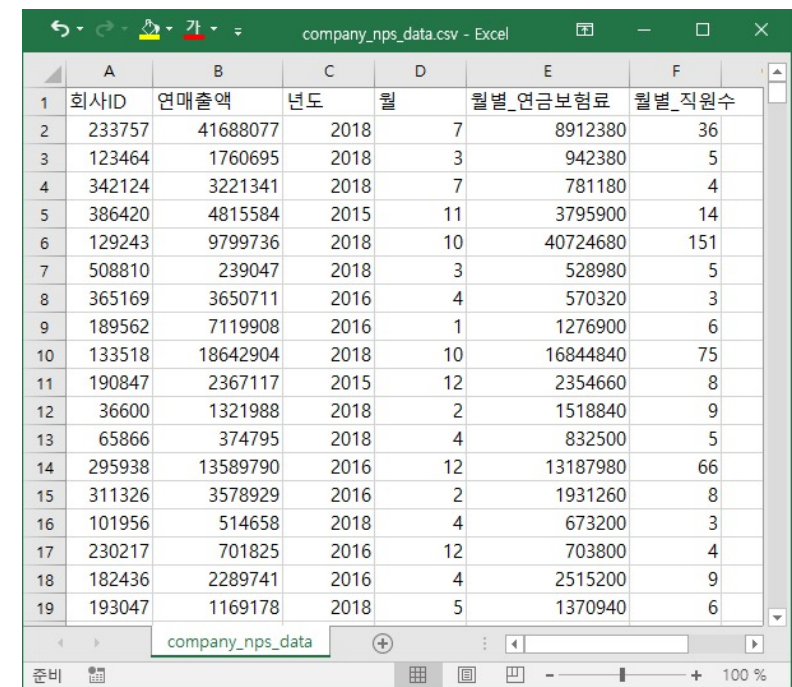
- 데이터 다운로드: <https://url.kr/jpn3bW>

■ 데이터 정보

- 회사 수: 약 5,000개
- 월별 데이터: 직원 수, 국민연금 보험료 (인원 수에 대한 상한선_최대고지금액 이 존재함)
- 년 단위 데이터: 매출액 (천원 단위)

■ EDA 목표

- 위 데이터들의 분포 (직원 수, 매출액, 보험료, +@)
- 몇몇 회사를 선택하여 데이터 흐름 살펴보기
- 데이터 사이의 관계에 대해 유의미한 결론을 찾아봅시다. (아이디어 수립 → EDA를 통해 관찰)



	A	B	C	D	E	F
	회사ID	연매출액	년도	월	월별_연금보험료	월별_직원수
1	회사ID	연매출액	년도	월	월별_연금보험료	월별_직원수
2	233757	41688077	2018	7	8912380	36
3	123464	1760695	2018	3	942380	5
4	342124	3221341	2018	7	781180	4
5	386420	4815584	2015	11	3795900	14
6	129243	9799736	2018	10	40724680	151
7	508810	239047	2018	3	528980	5
8	365169	3650711	2016	4	570320	3
9	189562	7119908	2016	1	1276900	6
10	133518	18642904	2018	10	16844840	75
11	190847	2367117	2015	12	2354660	8
12	36600	1321988	2018	2	1518840	9
13	65866	374795	2018	4	832500	5
14	295938	13589790	2016	12	13187980	66
15	311326	3578929	2016	2	1931260	8
16	101956	514658	2018	4	673200	3
17	230217	701825	2016	12	703800	4
18	182436	2289741	2016	4	2515200	9
19	193047	1169178	2018	5	1370940	6

팀프로젝트 진행방법

- 1) 팀장은 팀별로 main 저장소를 하나 생성합니다.
- 2) 각 팀원은 개별 작업내역은 main 저장소의 fork를 통해 작업합니다.
- 3) 분석 아이디어 및 협업은 main 저장소에 Issue를 통해 공유합니다.
- 4) 다음주 수업시작전에 팀원들간의 조율을 통해 작업내역을 main 저장소에 merge합니다.
- 5) 다음주에 팀별로 분석결과를 다른 사람들과 공유합니다.

각 팀별 Week1 과제 발표 시간

- 각 팀별로 Week1 과제에 대한 발표를 진행합니다.



Ch4. 머신러닝 & Scikit-learn

프리 온보딩 코스

에이아이스쿨(AISchool)

CONTENTS OF TABLES

1. 머신러닝(Machine Learning)

- 머신러닝(Machine Learning)이란?
- 머신러닝(Machine Learning)이 필요한 이유
- 예측 모델의 필요성
- 지도 학습(Supervised Learning)
- Training Data, Test Data
- 트레이닝 데이터, 테스트 데이터 나누기(split)
- 분류 문제(Classification) vs 회귀 문제(Regression)

2. 선형회귀(Linear Regression) & Scikit-learn

- 선형회귀(Linear Regression)
- 손실 함수(Loss Function) - MSE
- scikit-learn
- scikit-learn 기본 사용법
- scikit-learn을 training data, test data 나누기
- scikit-learn을 이용해서 MSE, RMSE 정의하기

3. 실제 예제로 예측 모델 만들어보기

- 연습문제 1. 보스턴 부동산 가격 예측해보기
- 보스턴 부동산 가격 데이터 특징들(Features)

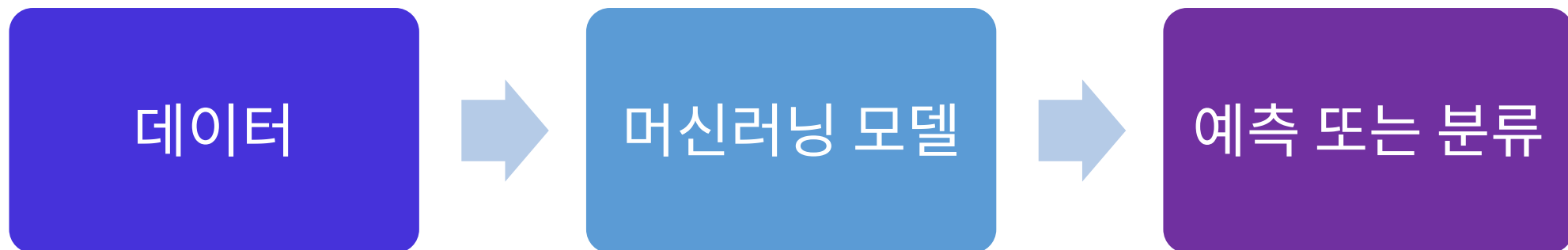
IV. 머신러닝 & Scikit-learn

코드 10줄로 시작하는 머신러닝



머신러닝(Machine Learning)이란?

- 머신러닝(Machine Learning)이란 명시적인 프로그래밍 없이 데이터를 이용해서 예측 또는 분류를 수행하는 알고리즘을 구현하는 기법을 뜻합니다.
- 머신러닝은 한국말로 기계 학습이라고도 부릅니다.



머신러닝(Machine Learning)이 필요한 이유

- 머신러닝 방법론을 이용할 경우, **인간이 정확히 하나하나 로직을 지정해주기 어려운 복잡한 문제**를 데이터에 기반한 학습을 통해서 해결할 수 있습니다.

e.g.) 어떤 사용자에게 무슨 광고를 보여주는 것이 최적의 광고 배분 전략일까?

- 머신러닝 알고리즘을 사용할때 가장 중요한 부분은 머신러닝 모델이 잘 학습할 수 있도록 **적절한 특징(Feature)**을 설정해주는 것 입니다.

예측 모델(Prediction Model)의 필요성

- 데이터 분석을 통한 정교한 **예측 모델(Prediction Model)**을 갖고 있을 경우, 중요한 비즈니스적 의사결정을 안정적이고 계획적으로 수행할 수 있습니다.

예제 1) 다음달에 휴대폰 판매량은 얼마나 될까? -> **생산 계획 및 재고관리 전략을 수립**할 수 있습니다.

예제 2) 광고비를 100만원 더 집행하면 얼마나 많은 유저를 추가적으로 획득할 수 있을까? -> 목표로 하는 유저 획득 수에 따른 **광고비 집행 전략**을 세울 수 있습니다.

지도 학습(Supervised Learning)

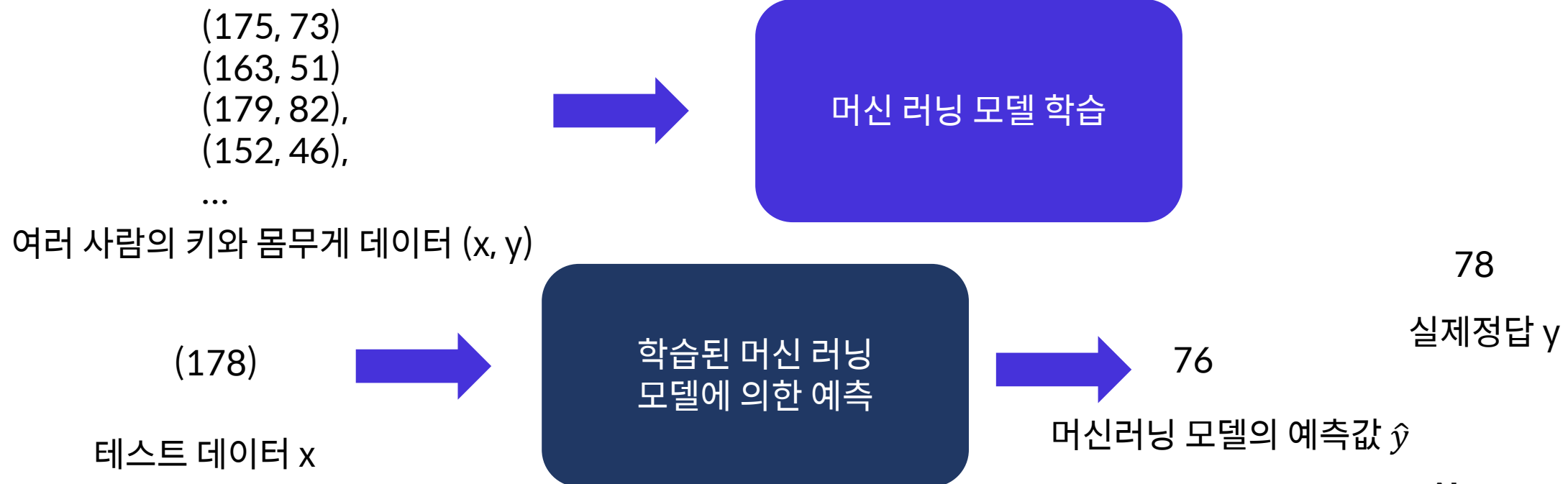
- 머신러닝 모델은 일반적으로 지도 학습(Supervised Learning)이라는 방법론을 사용합니다.
- 지도 학습 방법론을 사용하기 위해서는 트레이닝 데이터의 구성이 (인풋 데이터, 데이터에 대한 정답) 쌍으로 구성되어 있어야만 합니다. 즉, 지도 학습은 정답을 보여주면서 학습시키는 머신러닝 방법론입니다.
- 이때 보통 인풋 데이터를 x , 데이터에 대한 정답을 y 라고 부릅니다.
- 즉, 데이터는 (x, y) 쌍으로 구성됩니다.
- 예를 들어, 우리가 키를 기반으로 몸무게를 예측하는 모델을 만드는 경우를 생각해보면 트레이닝 데이터는 여러 사람에게서 수집한 키와 몸무게 데이터가 됩니다.

(175, 73)
(163, 51)
(179, 82),
(152, 46),
...

여러 사람의 키와 몸무게 데이터 (x, y)

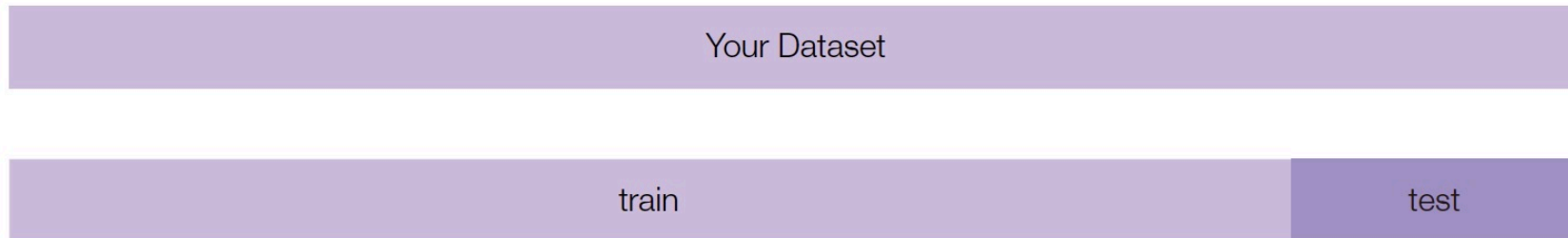
Training Data, Test Data

- 이런식으로 (x, y) 로 구성된 데이터를 **학습용 데이터(Training Data)**라고 부릅니다. 머신러닝 모델을 사용하는 경우 다음의 2가지 과정을 거칩니다.
- ① 학습용 데이터로 머신러닝 모델을 학습시킵니다.
- ② 학습된 머신러닝 모델의 성능을 트레이닝 데이터에 포함되어 있지 않고 따로 빼놓은 **테스트 데이터(Test Data)**로 측정합니다.



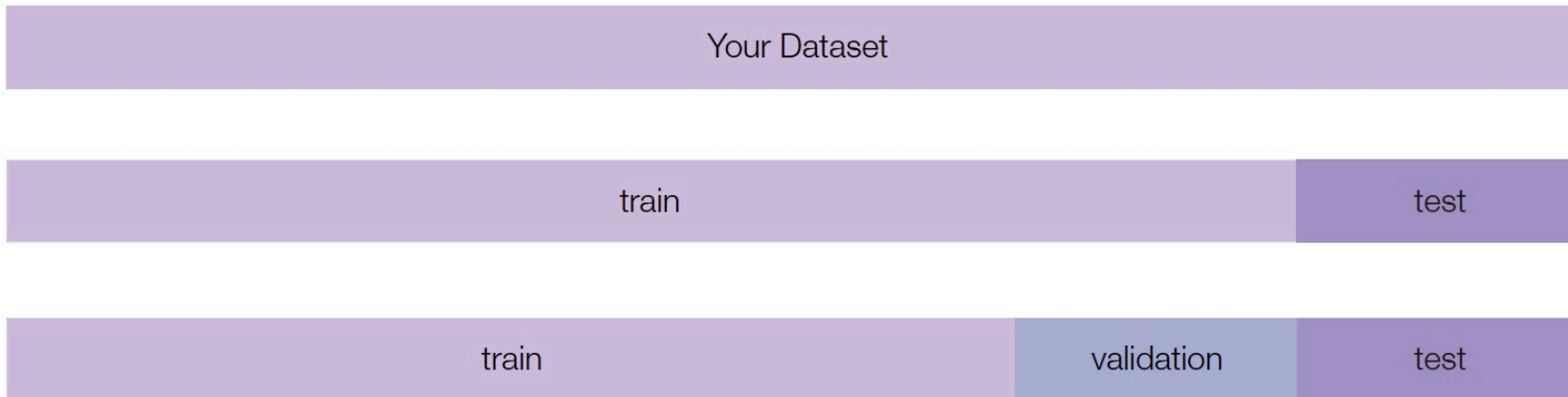
트레이닝 데이터, 테스트 데이터 나누기(split)

- 따라서 머신러닝 모델을 학습시키기 위해서 전체 데이터의 일부를 Training Data, 일부는 Test Data로 나눠서 사용합니다.
- 일반적으로 데이터의 **80%** 정도는 트레이닝 데이터, **20%** 정도는 테스트 데이터로 나눠서 사용합니다.
- 예를 들어 1000명의 (키, 몸무게) 데이터가 있다면 800명분의 데이터는 트레이닝 데이터, 200명분의 데이터는 테스트 데이터로 나눠서 사용합니다.



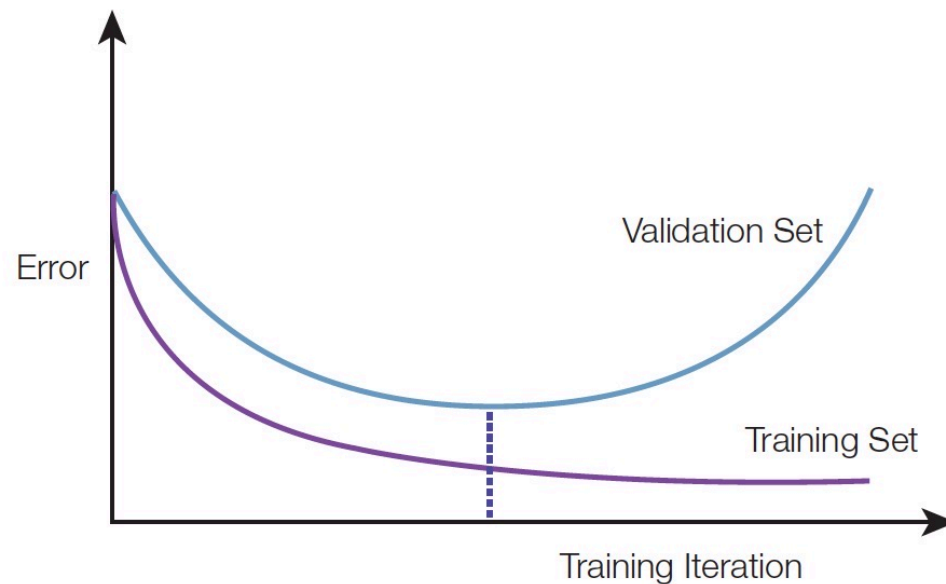
Validation Data(검증용 데이터)

- 여기에 더 나아가서 전체 데이터를 **트레이닝_{training} 데이터**, **검증용_{validation} 데이터**, **테스트_{test} 데이터**로 나누기도 합니다.
- 검증용 데이터는 트레이닝 과정에서 학습에 사용하지는 않지만 중간중간 테스트하는데 사용해서 학습하고 있는 모델이 오버피팅에 빠지지 않았는지 체크하는데 사용됩니다.
- 즉, 직관적으로 설명하면 검증용 데이터는 트레이닝 과정 중간에 사용하는 테스트 데이터로 볼 수 있습니다.



Overfitting, Underfitting

- 아래 그림은 학습 과정에서 트레이닝 에러와 검증 에러를 출력한 그래프입니다.
- 처음에는 트레이닝 에러와 검증 에러가 모두 작아지지만 일정 횟수 이상 반복할 경우 트레이닝 에러는 작아지지만 검증 에러는 커지는 **오버피팅(Overfitting)**에 빠지게 됩니다.
- 따라서 트레이닝 에러는 작아지지만 검증 에러는 커지는 지점에서 업데이트를 중지하면 최적의 파라미터를 얻을 수 있습니다.



Overfitting, Underfitting

- 오버피팅(Overfitting)은 학습 과정에서 머신러닝 알고리즘의 파라미터가 트레이닝 데이터에 과도하게 최적화되어 트레이닝 데이터에 대해서는 잘 동작하지만 새로운 데이터인 테스트 데이터에 대해서는 잘 동작하지 못하는 현상을 말합니다. 오버피팅은 모델의 표현력이 지나치게 강력할 경우 발생하기 쉽습니다.
- 그림 4-3은 오버피팅, 언더피팅의 경우를 보여줍니다. 그림 4-3의 가장 오른쪽 그림을 보면 모델이 트레이닝 데이터의 정확도를 높이기 위해 결정 직선(Decision Boundary)을 과도하게 꼬아서 그린 모습을 볼 수 있습니다. 이런 경우 트레이닝 데이터와 조금 형태가 다른 새로운 데이터를 예측할 때도 성능이 떨어지게 됩니다.
- 이에 반해 그림 4-3의 가장 왼쪽 그림은 언더피팅(Underfitting)에 빠진 상황을 보여줍니다. 언더피팅은 오버피팅의 반대 상황으로 모델의 표현력이 부족해서 트레이닝 데이터도 제대로 예측하지 못하는 상황을 말합니다. 마지막으로 그림 4-3의 중앙에 있는 그림은 오버피팅과 언더피팅에 빠지지 않고 파라미터가 적절히 학습된 경우를 보여줍니다.
- 이런 오버피팅을 방지하기 위한 기법들을 Regularization 기법이라고 부릅니다.

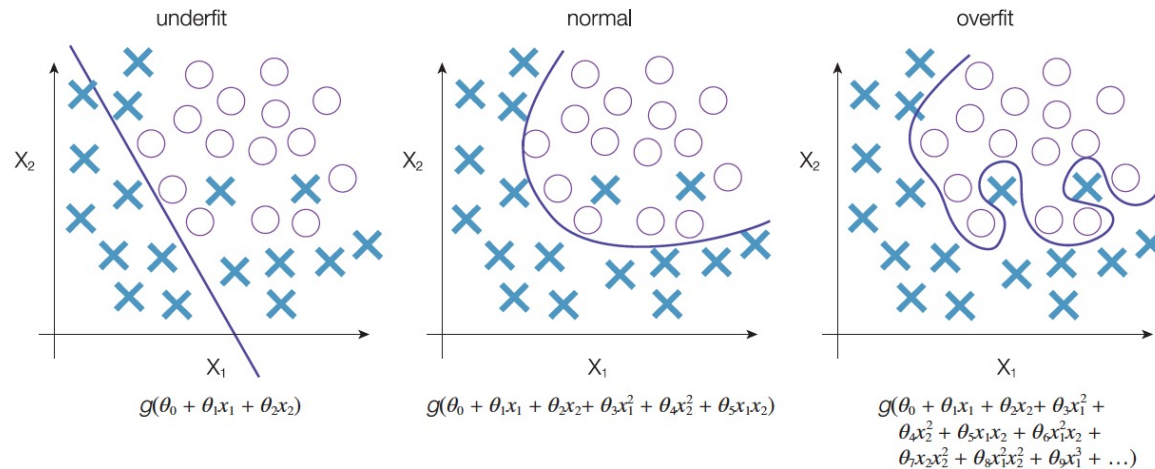


그림 4-3 | 언더피팅, 적절히 학습된 경우, 오버피팅

분류 문제(Classification) vs 회귀 문제(Regression)

- 이때 예측 모델 값의 형태에 따라서 분류 문제 혹은 회귀 문제로 나뉩니다.

- **분류** Classification 문제 : 예측하는 결과값이 이산값 Discrete Value 인 문제

e.g. 이 이미지에 해당하는 숫자는 1인가 2인가?

- **회귀** Regression 문제 : 예측하는 결과값이 연속값 Continuous Value 인 문제

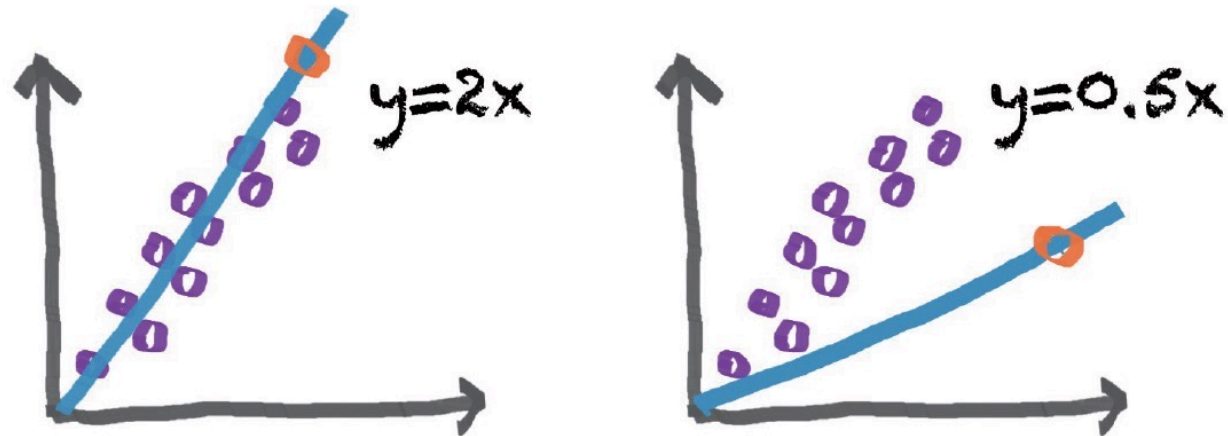
e.g. 3개월 뒤 이 아파트 가격은 2억1천만 원일 것인가? 2억2천만 원일 것인가?

선형회귀(Linear Regression)

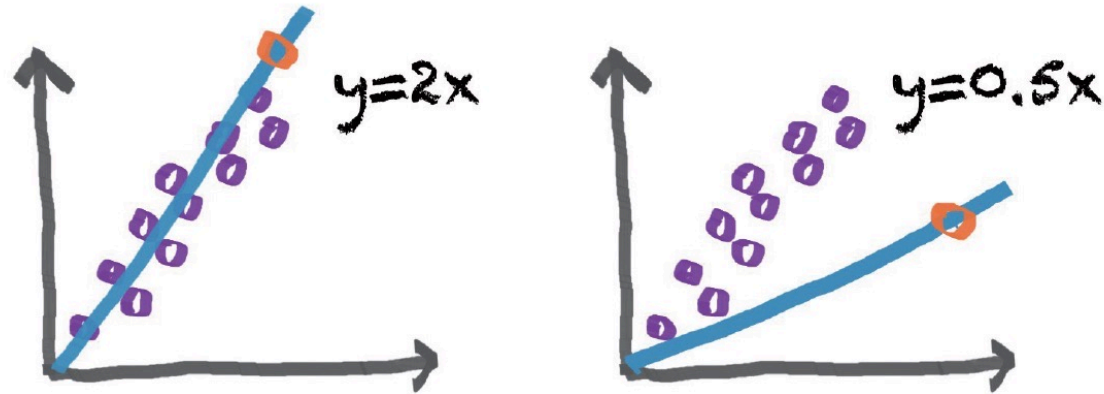
- 선형 회귀(Linear Regression) 모델은 아래와 같은 선형 함수를 이용해서 회귀(Regression)를 수행하는 모델을 뜻합니다.

$$y = Wx + b$$

- 이때 x, y 는 저희가 가지고 있는 데이터이고, w 와 b 는 데이터에 적합한 값으로 학습될 수 있는 **파라미터 (Parameter)** 입니다.



선형회귀(Linear Regression) 예제



- 보라색 동그라미는 트레이닝 데이터, 파란색 라인은 선형 회귀 기법이 학습한 가설, 주황색 동그라미는 학습한 가설을 바탕으로 테스트 데이터에 대해 예측을 수행한 결과입니다.
- 왼쪽 그림은 선형 회귀 모델이 $y=2x$ ($W=2, b=0$)로 가설을 학습한 경우, 오른쪽 그림은 선형 회귀 모델이 $y=0.5x$ ($W=0.5, b=0$)로 가설을 학습한 경우입니다.
- 그림에서 볼 수 있듯이 보라색의 트레이닝 데이터는 **$y=2x$ 형태의 경향성**을 띠고 있기 때문에 선형 회귀 모델이 잘 학습된 경우 $y=2x$ 형태의 가설을 가지고 있어야 합니다.
- 만약 잘못된 선형 함수가 학습된 경우 오른쪽 그림과 같이 테스트 데이터에 대해 부정확한 예측값을 출력하게 될 것입니다.

손실 함수(Loss Function) - MSE

- 머신 러닝 모델을 학습시키기 위해서는 적절한 파라미터값을 알아내기 위해서 현재 파라미터값이 우리가 풀고자 하는 목적_{Task}에 적합한 값인지를 측정할 수 있어야 합니다. 이를 위해 **손실 함수** Loss Function **J(θ)**를 정의합니다.
- 손실 함수는 여러가지 형태로 정의될 수 있습니다. 그 중 가장 대표적인 손실 함수 중 하나는 **평균제곱오차** Mean of Squared Error(MSE)입니다.
- MSE는 다음 수식으로 정의됩니다.

$$MSE = \frac{1}{2n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$$

손실 함수(Loss Function) - MSE

- 예를 들어, 정답이 $y=[1, 10, 13, 7]$ 이고 우리의 모델의 예측값이 $\hat{y}=[10, 3, 1, 4]$ 와 같이 잘못된 값을 예측한다면 MSE 손실 함수는 아래와 같이 35.375라는 큰 값을 갖게 될 것입니다.

$$MSE = \frac{1}{2 * 4} \{(10 - 1)^2 + (3 - 10)^2 + (1 - 13)^2 + (4 - 7)^2\} = 35.375$$

- 하지만, 정답이 $y=[1, 10, 13, 7]$ 이고 우리의 모델의 예측값이 $\hat{y}=[2, 10, 11, 6]$ 와 같이 비슷한 값을 예측한다면 MSE 손실 함수는 아래와 같이 1.5라는 작은 값을 갖게 될 것입니다.

$$MSE = \frac{1}{2 * 4} \{(2 - 1)^2 + (10 - 10)^2 + (11 - 13)^2 + (6 - 7)^2\} = 1.5$$

- 이처럼 손실 함수는 우리가 풀고자 하는 목적에 가까운 형태로 파라미터가 최적화 되었을 때(즉, 모델이 잘 학습되었을 때) 더 작은 값을 갖는 특성을 가져야만 합니다.
- 이런 특징 때문에 손실 함수를 다른 말로 **비용 함수** Cost Function 라고도 부릅니다.

scikit-learn

- 방금 살펴본 선형 회귀(Linear Regression)을 포함한 다양한 머신러닝 모델을 쉽고 간편하게 구현할 수 있도록 도와주는 라이브러리가 **scikit-learn**입니다.
- <https://scikit-learn.org/>

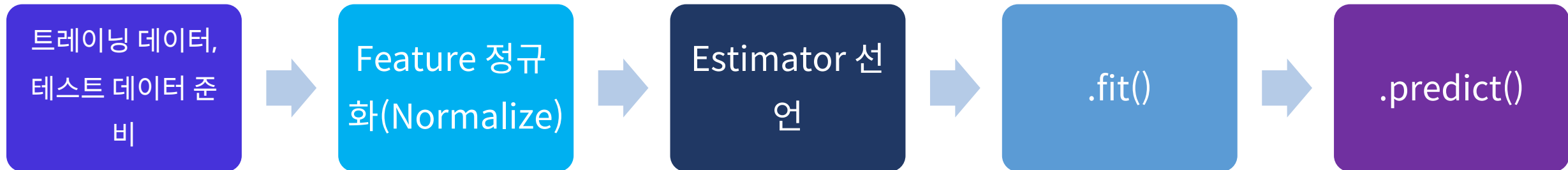


scikit-learn 기본 사용법

- scikit-learn의 기본 사용법은 다음과 같습니다.
 - ① Estimator 선언 (e.g. LinearRegression)
 - ② .fit() 함수 호출을 통한 트레이닝 (Training)
 - ③ .predict() 함수 호출을 통한 예측 (Predict)

scikit-learn을 이용한 예측 모델 구성방법

- scikit-learn을 이용해서 예측모델을 생성하는 방법은 다음과 같습니다.



scikit-learn을 training data, test data 나누기

- scikit-learn을 이용해서 training data, test data를 나누는 방법은 다음과 같습니다.

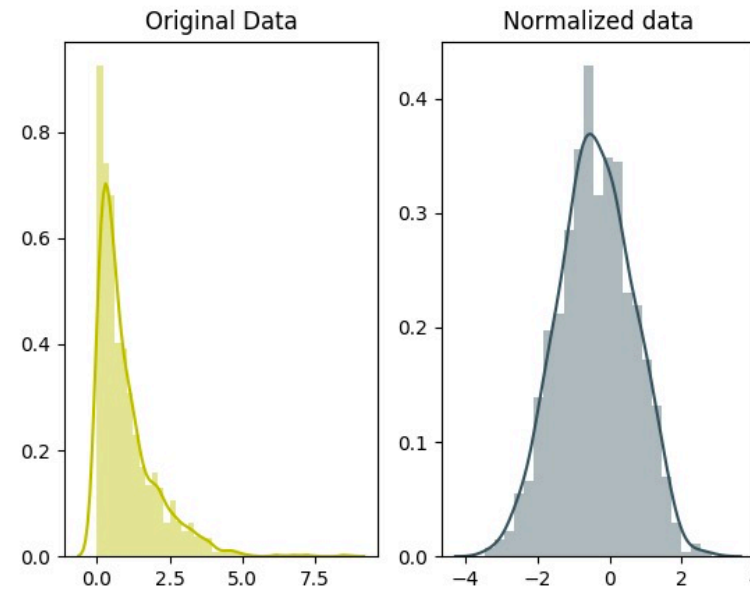
```
from sklearn.model_selection import train_test_split

# 80%는 트레이닝 데이터, 20%는 테스트 데이터로 나눕니다.
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

Feature 정규화(Normalize)하기

- Feature를 정규화할 경우, 값의 분포를 정규 분포(Normal Distribution) 형태로 변경할 수 있습니다.
- 일반적으로 Feature 값에 대한 정규화를 수행할 경우, 더 안정적으로 머신러닝 모델을 학습시킬 수 있습니다.

$$\frac{X - \mu}{\sigma}$$



이미지 출처 :

<https://kharshit.github.io/blog/2018/03/23/scaling-vs-normalization>

Feature 정규화(Normalize)하기

- StandardScaler 클래스를 이용해서 Feature를 정규화 할 수 있습니다.

```
from sklearn import preprocessing  
  
X = boston_housde_df.iloc[:, :-1]  
# StandardScaler를 이용해서 데이터 정규화(Noramlize)하기  
X = preprocessing.StandardScaler().fit(X).transform(X)
```

scikit-learn을 선형회귀(Linear Regression) Estimator 선언하기

- Scikit-learn을 이용해서 선형회귀(Linear Regression) Estimator를 선언하는 방법은 다음과 같습니다.

```
from sklearn.linear_model import LinearRegression  
  
# 선형회귀(Linear Regression) 모델 선언하기  
lr = LinearRegression()
```

scikit-learn을 이용해서 MSE, RMSE 정의하기

- MSE는 차이를 제공하기 때문에 제공에 의해서 생기는 오차를 보정하기 위해서 RMSE(Root Mean Square Error)를 이용해서 성능을 측정하기도 합니다.
- scikit-learn에서 MSE와 RMSE를 정의하는 방법은 다음과 같습니다.

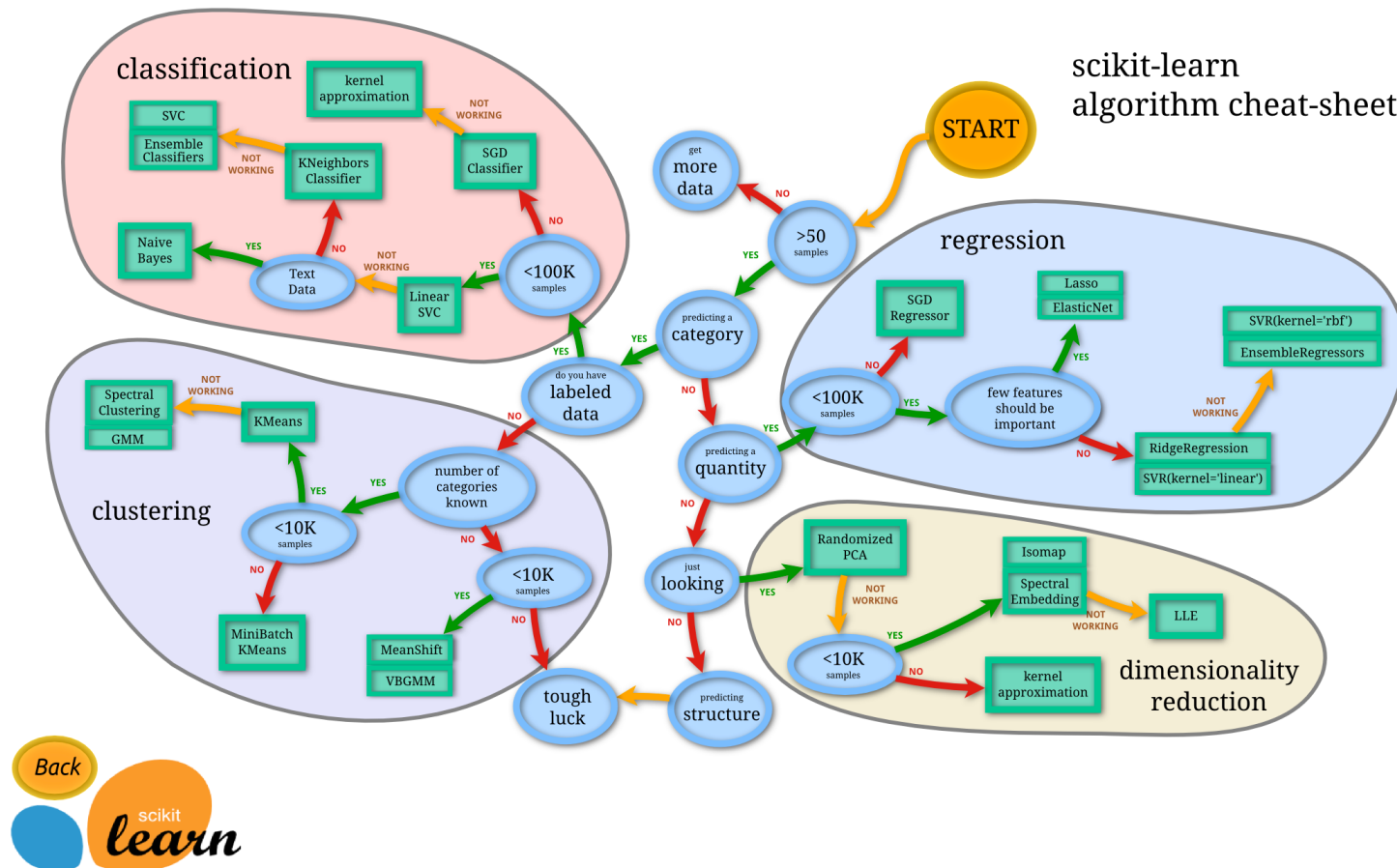
```
from sklearn.metrics import mean_squared_error

# MSE(Mean Squared Error)를 측정합니다.
MSE = mean_squared_error(y_test, y_preds)

# RMSE(Root Mean Squared Error)를 측정합니다.
RMSE = np.sqrt(MSE)
```

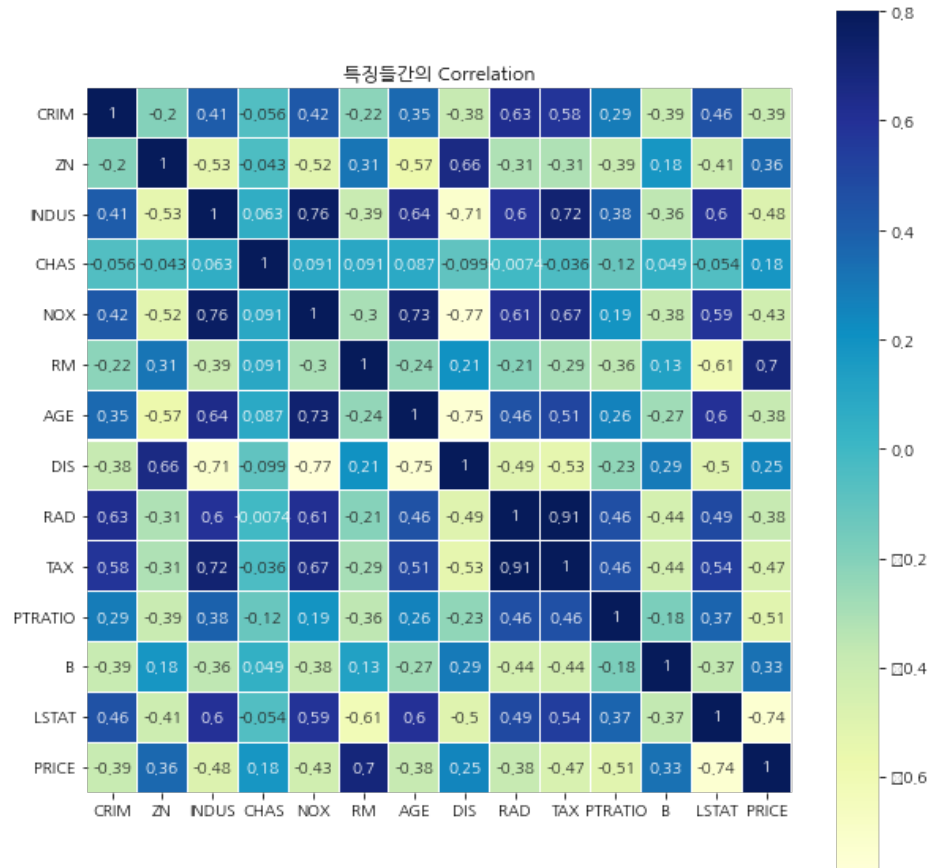
scikit-learn을 이용해서 상황에 따른 적절한 모델(Estimator) 선택하기

- https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html



상관 분석(Correlation Analysis)

- **상관 분석(Correlation analysis)** 또는 '상관관계' 또는 '상관'은 확률론과 통계학에서 두 변수간에 어떤 선형적 또는 비선형적 관계를 갖고 있는지를 분석하는 방법이다.

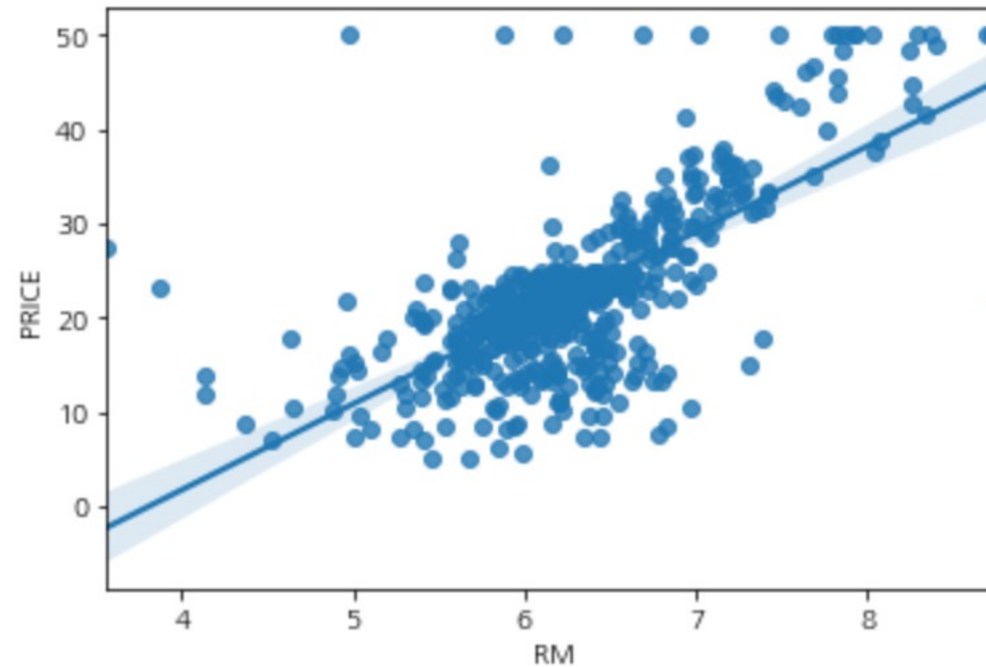


```
corr = boston_housde_df.corr()
plt.figure(figsize=(10, 10));
sns.heatmap(corr,
             vmax=0.8,
             linewidths=0.01,
             square=True,
             annot=True,
             cmap='YlGnBu');
plt.title('특징들간의 Correlation');
```


sns.regplot으로 Feature들 간의 경향성 출력해보기

- `sns.regplot(data={dataframe}, x={컬럼명}, y={컬럼명})` 형태를 이용해서 regression line이 포함된 scatter plot을 그릴 수 있습니다.

```
sns.regplot(data=boston_housde_df, x='RM', y='PRICE')  
plt.show()
```



연습문제 1. 보스턴 부동산 가격 예측해보기

- <https://www.kaggle.com/vikrishnan/boston-house-prices>
- 실습 데이터는 1970년도의 보스턴 지역의 집값을 나타냅니다.
- 선형 회귀(Linear Regression) 알고리즘을 이용해서 보스턴 부동산 가격을 예측해봅시다!



보스턴 부동산 가격 데이터 특징들(Features)

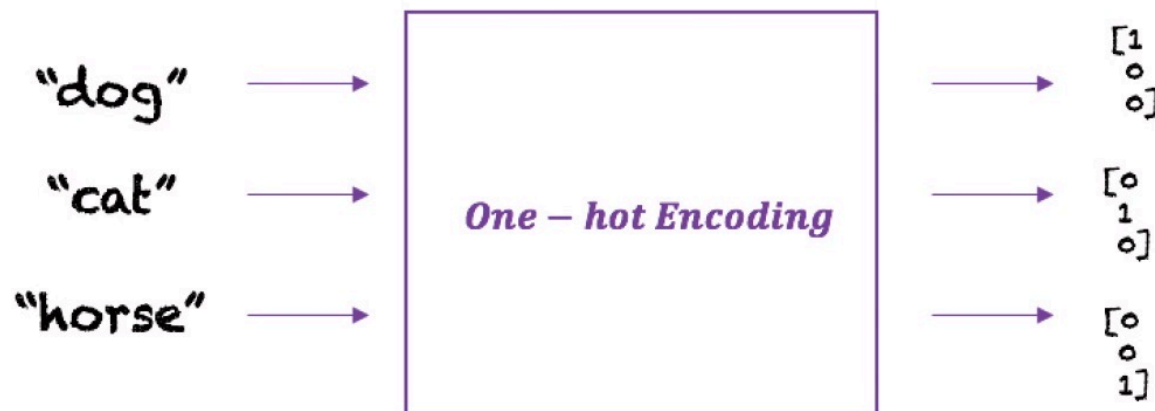
- ① CRIM: 도시별 범죄발생률
- ② ZN: 25,000평을 넘는 토지의 비율
- ③ INDUS: 도시별 비상업 지구의 비유
- ④ CHAS: 찰스 강의 더미 변수(1 = 강의 경계, 0 = 나머지)
- ⑤ NOX: 일산화질소 농도
- ⑥ RM: 주거할 수 있는 평균 방의개수
- ⑦ AGE: 1940년 이전에 지어진 주택의 비율
- ⑧ DIS: 5개의 고용지원센터까지의 가중치가 고려된 거리
- ⑨ RAD: 고속도로의 접근 용이성에 대한 지표
- ⑩ TAX: 10,000달러당 재산세 비율
- ⑪ PTRATIO: 도시별 교사와 학생의 비율
- ⑫ B: 도시의 흑인 거주 비유
- ⑬ LSTAT: 저소득층의 비율
- ⑭ MEDV: 본인 소유 주택 가격의 중앙값

연습문제 1. 보스턴 부동산 가격 예측해보기

<4강_연습문제1_부동산가격_예측해보기.ipynb>

One-hot Encoding

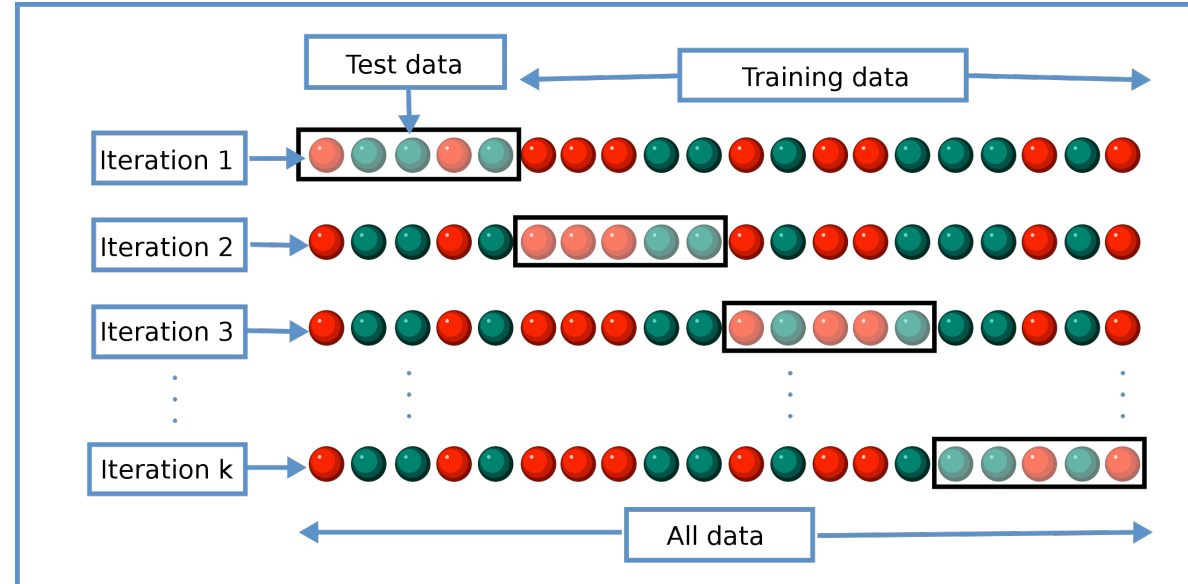
- **One-hot Encoding**은 범주형 값 Categorical Value을 이진화된 값 Binary Value으로 바꿔서 표현하는 것을 의미합니다. 범주형 값은 예를 들어 “개”, “고양이”, “말”이라는 3개의 범주형 데이터가 있을 때 이를 [“개”=1, “고양이”=2, “말”=3]이라고 단순히 Integer Encoding으로 변환하여 표현하는 것입니다.
- 이에 반해 One-hot Encoding을 사용하면 범주형 데이터를 “개”=[1 0 0], “고양이”=[0 1 0], “말”=[0 0 1] 형태로 해당 레이블을 나타내는 인덱스만 1의 값을 가지고 나머지 부분은 0의 값을 가진 Binary Value로 표현합니다.



- 단순한 Integer Encoding의 문제점은 머신러닝 알고리즘이 정수 값으로부터 잘못된 경향성을 학습하게 될 수도 있다는 점입니다. 예를 들어, 위의 예시의 경우 Integer Encoding을 사용할 경우 머신러닝 알고리즘이 [“개”(=1)와 “말”(=3)의 평균($1+3/2=2$)은 “고양이”(=2)이다.] 라는 지식을 학습할 수도 있는데, 이는 명백히 잘못된 학습입니다. 따라서 머신 러닝 알고리즘을 구현할 때 타겟 데이터를 One-hot Encoding 형태로 표현하는 것이 일반적입니다.

K-Fold Cross Validation (K-Fold 교차 검증)

- [https://en.wikipedia.org/wiki/Cross-validation_\(statistics\)](https://en.wikipedia.org/wiki/Cross-validation_(statistics))
- 데이터의 개수가 너무 작을 경우, 트레이닝 데이터와 테스트 데이터가 어떻게 나뉘지는가에 따라 학습된 모델과 성능 측정결과가 크게 달라질 수 있습니다.
- 따라서 이러 문제를 해결하기 위해 K-Fold Cross Validation (K-Fold 교차 검증)을 사용할 수 있습니다.



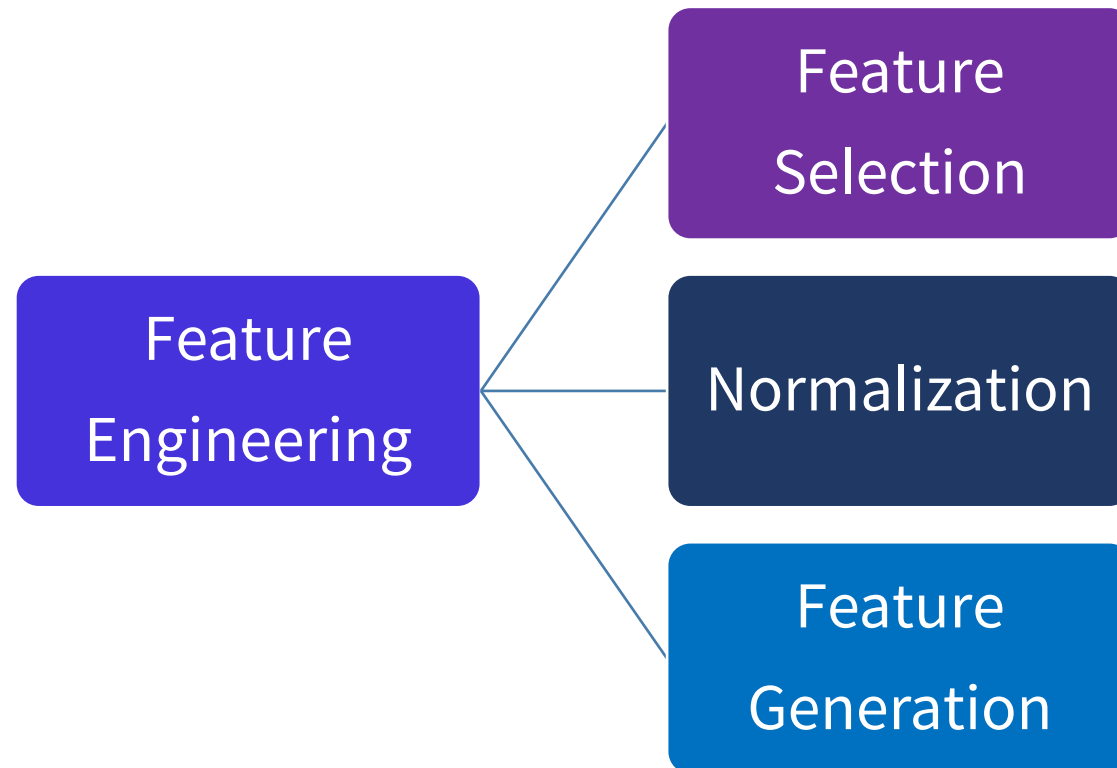
scikit-learn을 K-Fold Cross Validation 구현하기

- scikit-learn을 이용해서 K-Fold Cross Validation을 구현하는 방법은 다음과 같습니다.
- https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html

```
>>> import numpy as np
>>> from sklearn.model_selection import KFold
>>> X = np.array([[1, 2], [3, 4], [1, 2], [3, 4]])
>>> y = np.array([1, 2, 3, 4])
>>> kf = KFold(n_splits=2)
>>> kf.get_n_splits(X)
2
>>> print(kf)
KFold(n_splits=2, random_state=None, shuffle=False)
>>> for train_index, test_index in kf.split(X):
...     print("TRAIN:", train_index, "TEST:", test_index)
...     X_train, X_test = X[train_index], X[test_index]
...     y_train, y_test = y[train_index], y[test_index]
TRAIN: [2 3] TEST: [0 1]
TRAIN: [0 1] TEST: [2 3]
```

Feature Engineering

- **Feature Engineering**은 도메인 지식이나 분석을 통해서 유의미한 특징(Feature)들만을 선별해내거나 Feature의 형태를 더욱 적합한 형태로 변경하는 것을 뜻합니다.
- 적절한 Feature Engineering 머신러닝 모델의 성능에 큰 영향을 끼칠 수 있습니다.



Feature Selection

- **Feature Selection**은 예측값과 연관이 없는 불필요한 특징을 제거해서 머신러닝 모델의 성능을 더욱 높이는 기법입니다.
- Feature Selection에서 제거할 특징을 선택하기 위해 **상관 분석(Correlation Analysis)** 등을 진행할 수 있습니다.

Feature
Selection



Step Up! 오늘 배운 내용 마스터하기

- scikit-learn 공식 홈페이지, tutorial

<https://scikit-learn.org/stable/#>

- Linear Regression 알고리즘 설명

https://ko.wikipedia.org/wiki/%EC%84%A0%ED%98%95_%ED%9A%8C%EA%B7%80

- scikit-learn cheatsheet

https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Scikit_Learn_Cheat_Sheet_Python.pdf

Q & A



THANK YOU :)

Ch5. 수요 예측 프로젝트

프리 온보딩 코스

에이아이스쿨(AISchool)

CONTENTS OF TABLES

1. 데이터 정제

- 데이터 클렌징(Data Cleansing)
- Feature Engineering
- 상관 분석(Correlation Analysis)
- DataFrame 추가 method 학습하기 - unique, replace, filtering

2. 광고 캠페인 분석해보기

- 광고 캠페인의 기본 개념과 용어 이해하기
- 페이스북 광고 캠페인 분석해보기
- 페이스북 광고 캠페인의 특징들
- 실전 예제 1. 광고 캠페인 전환율 및 성과 분석하기

3. 랜덤 포레스트(Random Forest)

- 결정 트리(Decision Tree)
- 랜덤 포레스트(Random Forest)
- 앙상블 러닝(Ensemble Learning)
- scikit-learn의 Random Forest Estimator
- 랜덤 포레스트의 하이퍼 파라미터

4. 수요 예측 진행해보기

- 월마트 상품 판매량 판매량 수요예측 문제 정의
- 성능 평가 방법
- 데이터 설명(Data Description)
- 실전 예제 2. 대형마트의 상품 판매량 예측 프로젝트 (월마트, 매장별로 내일 얼마 나 팔릴까?)

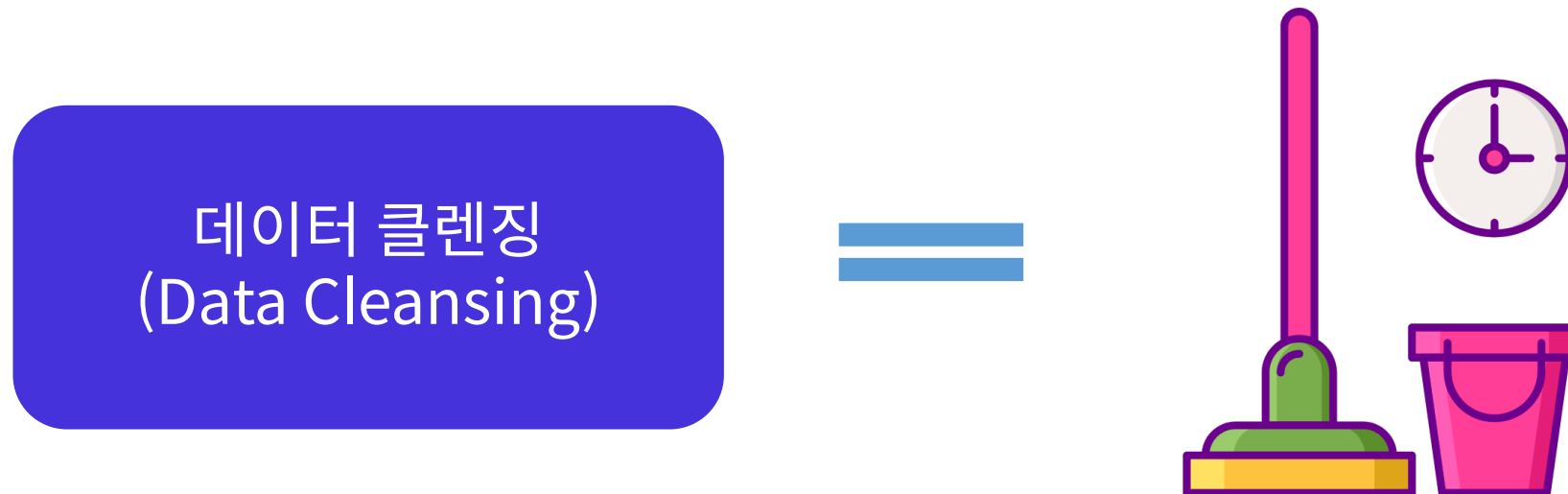
V. 수요 예측 프로젝트

취업 시장에서 가장 원하는 능력인 데이터 분석의 꽃,
마케팅 데이터분석 및 수요 예측 프로젝트 시작하기



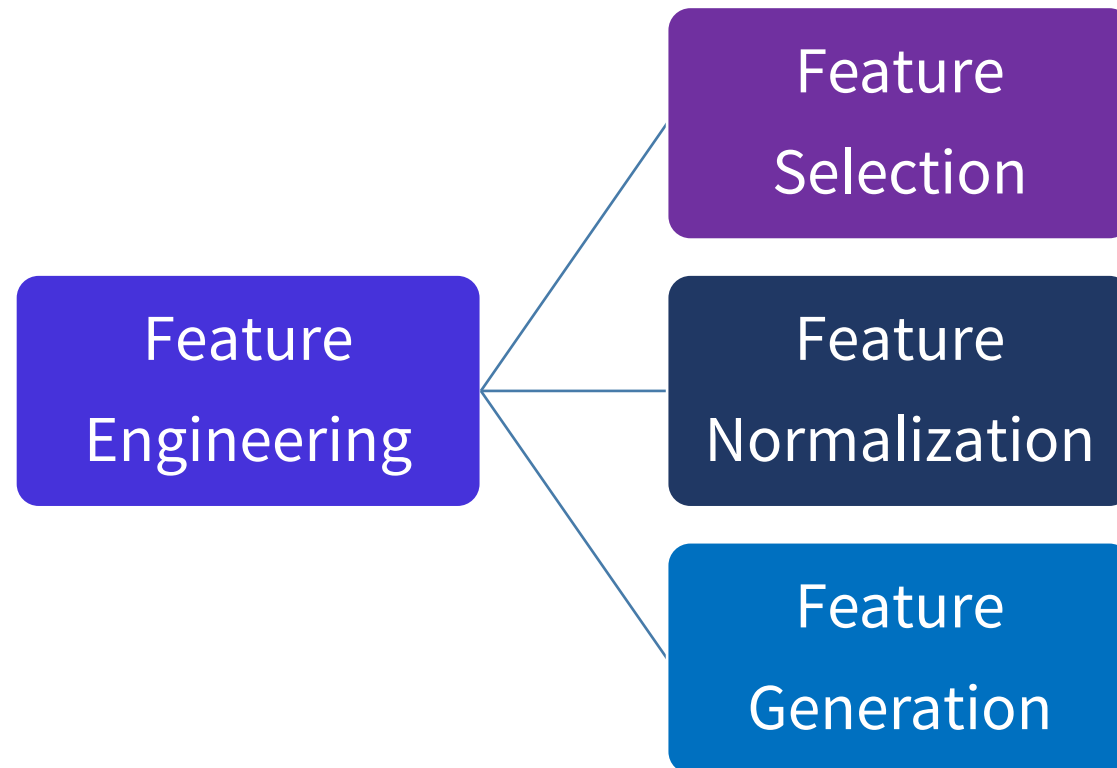
데이터 클렌징(Data Cleansing) – 데이터에서 유의미한 내용들만 골라내기 (데이터 청소하기)

- 데이터 클렌징(Data Cleansing)은 데이터에 존재하는 이상치(outlier)나 결측치(Missing value)를 제거해서 데이터를 정리하는 것을 의미합니다.
- 즉, 데이터에서 불필요한 부분을 청소해내는 과정이라고 생각할 수 있습니다.



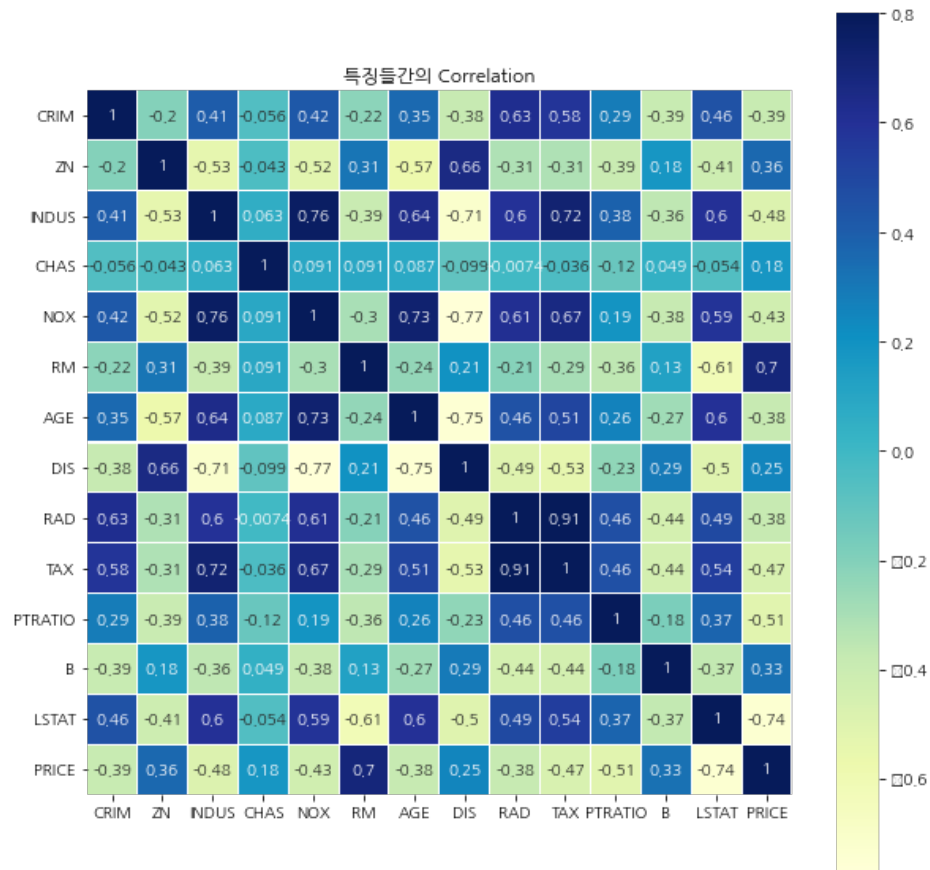
Feature Engineering

- **Feature Engineering**은 도메인 지식이나 분석을 통해서 유의미한 특징(Feature)들만을 선별해내거나 Feature의 형태를 더욱 적합한 형태로 변경하는 것을 뜻합니다.
- 적절한 Feature Engineering 머신러닝 모델의 성능에 큰 영향을 끼칠 수 있습니다.



상관 분석(Correlation Analysis)

- **상관 분석(Correlation analysis)** 또는 '상관관계' 또는 '상관'은 확률론과 통계학에서 두 변수간에 어떤 선형적 또는 비선형적 관계를 갖고 있는지를 분석하는 방법이다.



```
corr = boston_housde_df.corr()
plt.figure(figsize=(10, 10));
sns.heatmap(corr,
             vmax=0.8,
             linewidths=0.01,
             square=True,
             annot=True,
             cmap='YlGnBu');
plt.title('특징들간의 Correlation');
```

DataFrame 추가 method 학습하기 – unique, replace, filtering

- ① DataFrame의 특정 Column의 unique한 값들 얻기
 - {DataFrame변수명}.unique()
- ② DataFrame의 일부 값 수정하기
 - {DataFrame변수명}.replace({{원래값}:{변경할값}}, inplace=True)

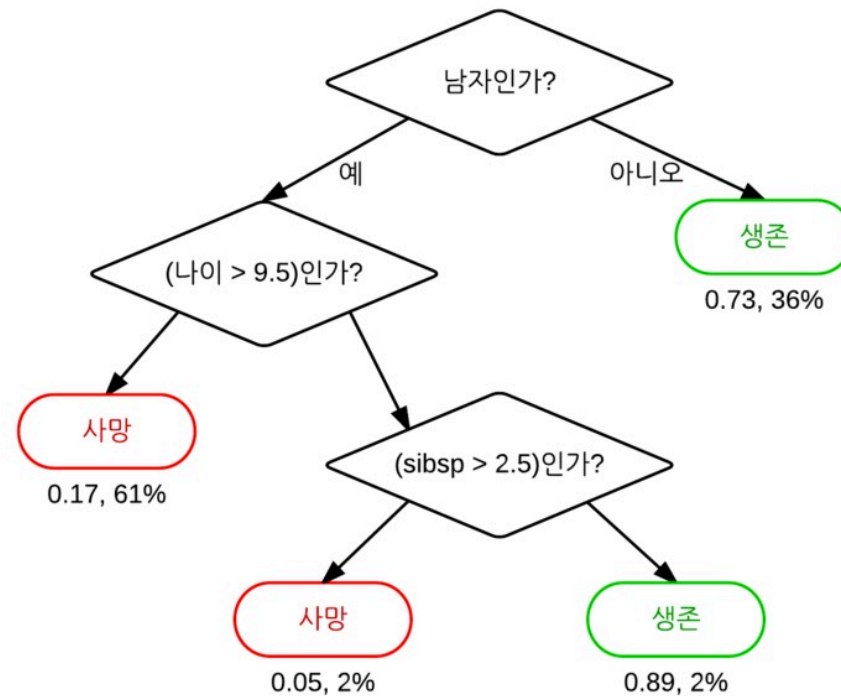
```
df["xyz_campaign_id"].replace({916:"campaignA",936:"campaignB",1178:"campaignC"},inplace=True)
```

- ③ DataFrame 필터링후 일부 조건을 만족하는 새 DataFrame 만들기
 - {새로만들 DataFrame} = {기존 DataFrame변수명}[기존 DataFrame변수명['{컬럼명}']=={조건}]

```
campaign_c = df[df['xyz_campaign_id']=='campaignC']
```

결정 트리(Decision Tree)

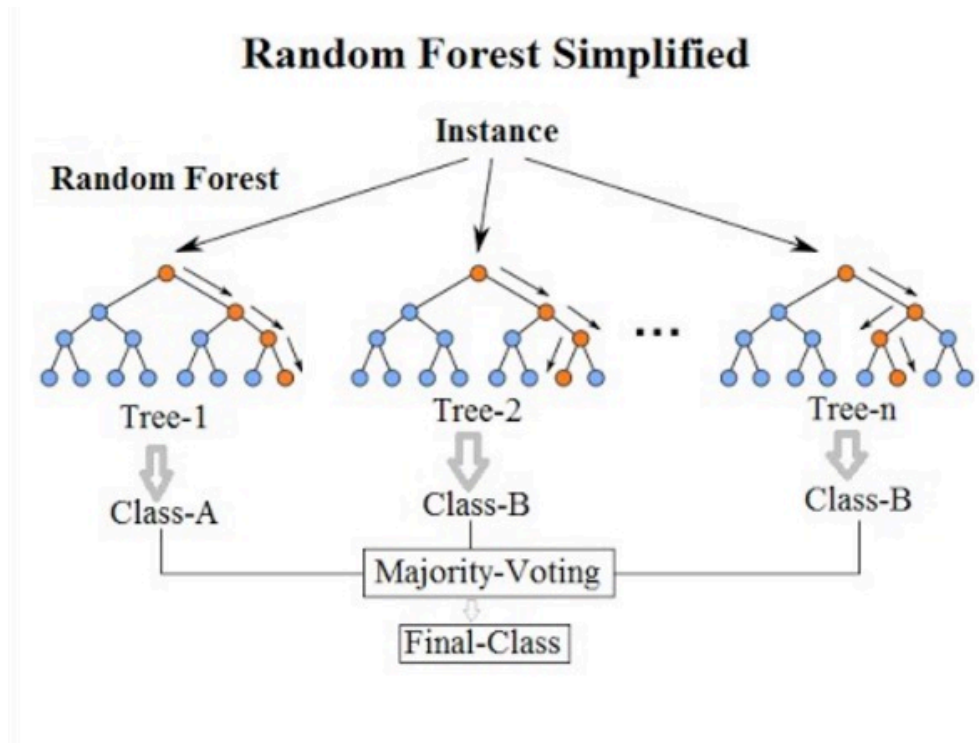
- 결정 트리(Decision Tree) 학습법은 데이터 마이닝에서 일반적으로 사용되는 방법론으로, 몇몇 입력 변수를 바탕으로 목표 변수의 값을 예측하는 모델을 생성하는 것을 목표로 한다.
- 아래 그림은 그러한 예측 모델의 한 예를 나타내고 있다. 그림의 트리 구조에서, 각 내부 노드들은 하나의 입력 변수에, 자녀 노드들로 이어지는 가지들은 입력 변수의 가능한 값에 대응된다.
- 잎 노드는 각 입력 변수들이 루트 노드로부터 잎 노드로 이어지는 경로에 해당되는 값들을 가질 때의 목표 변수 값에 해당된다.



자료 출처:
https://ko.wikipedia.org/wiki/%EA%B2%B0%EC%A0%95_%ED%8A%B8%EB%A6%AC_%ED%95%99%EC%8A%B5%EB%B2%95

랜덤 포레스트(Random Forest)

- **랜덤 포레스트(영어: random forest)**는 분류, 회귀 분석 등에 사용되는 앙상블 학습 방법의 일종으로, 훈련 과정에서 구성된 다수의 결정 트리로부터 부류(분류) 또는 평균 예측치(회귀 분석)를 출력함으로써 동작한다.
- 랜덤포레스트는 전체 데이터의 일부를 샘플링한 서브 데이터를 이용해서 학습시킨 **여러개의 결정트리의 예측값들간에 보팅을 통해서 최종 출력값을 만들어내는 기법**입니다.



자료 출처 :

https://en.wikipedia.org/wiki/Random_forest
<https://ko.wikipedia.org/wiki/%EB%9E%9C%EB%8D%A4%ED%8F%AC%EB%A0%88%EC%8A%A4%ED%8A%B8>

앙상블 러닝(Ensemble Learning)

- 앙상블 러닝은 여러 개의 분류기의 예측 결과값 간의 투표를 통해서 최종 결과값을 만들어내는 기법입니다.
- 앙상블 러닝을 이용할 경우, 더욱 좋은 예측 성능을 기대할 수 있습니다.

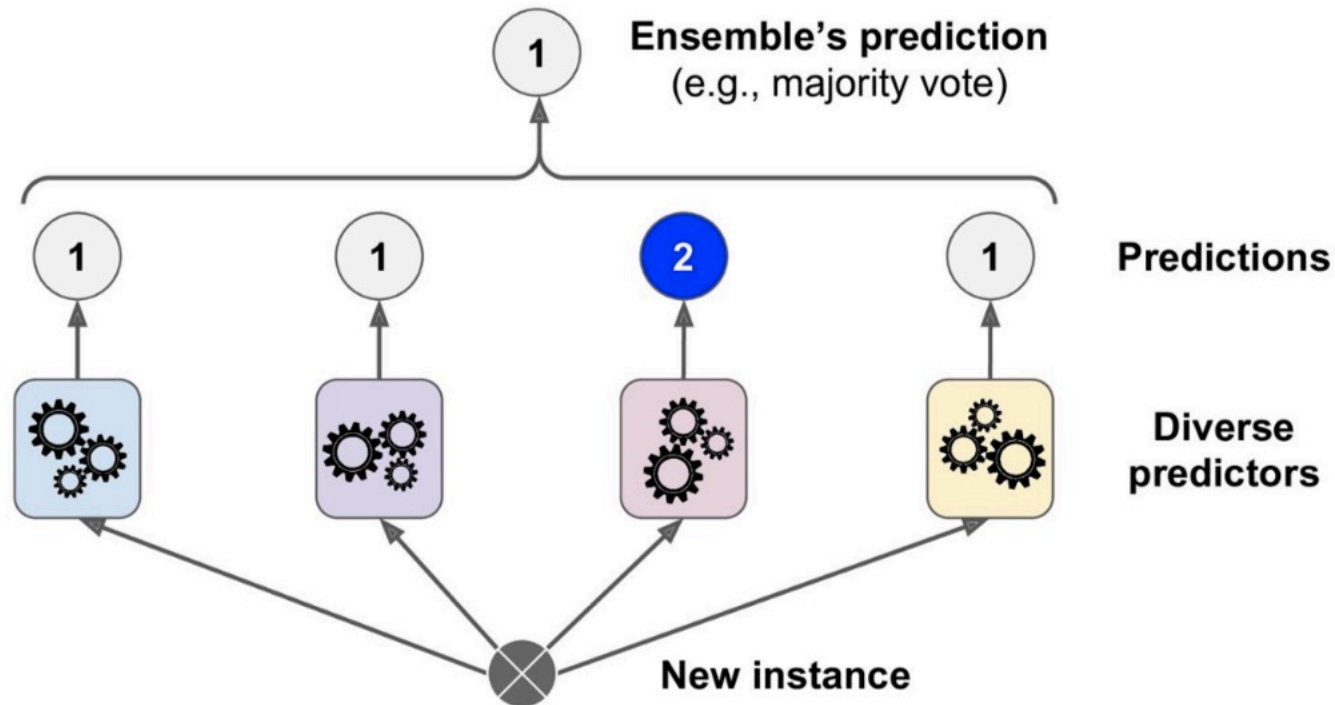


Figure 7-2. Hard voting classifier predictions

scikit-learn의 Random Forest Estimator

- scikit-learn에서 랜더 포레스트를 구현한 Estimator는 다음과 같이 2가지입니다.
- `sklearn.ensemble.RandomForestClassifier` (분류(Classification) 문제에 사용)
- `sklearn.ensemble.RandomForestRegressor` (회귀(Regression) 문제에 사용)

랜덤 포레스트의 하이퍼 파라미터

- 하이퍼 파라미터(hyper-parameter)란 알고리즘의 동작 과정에 영향을 미치는 중요한 값들로서 알고리즘 디자이너가 결정해줘야만 하는 값들입니다.
- 랜덤포레스트의 하이퍼 파라미터는 다음과 같습니다.
 - ① **n_estimators** : 랜덤 포레스트에서 사용할 결정트리 개수를 지칭합니다. 기본값은 100개입니다. 많이 설정할 수록 성능이 향상될 수 있지만 학습 시간이 오래걸릴 수 있습니다.
 - ② **max_features** : 결정트리 분할 기준으로 사용할 Feature 개수
 - ③ **max_depth** : 트리의 최대 깊이, 너무 깊어지면 오버피팅이 발생할 가능성이 있음.
 - ④ **min_samples_split** : 노드를 분할하기 위한 최소한의 샘플 데이터수, 너무 작은 경우 과적합이 발생할 가능성이 높아집니다. 기본값은 2입니다.

실전 예제 2. 대형마트의 상품 판매량 예측 프로젝트 (월마트, 매장별로 내일 얼마나 팔릴까?)

- 월마트에서 데이터 사이언티스트를 채용하기 위해 올린 Kaggle Competition입니다.
- 월마트의 매점별 판매량 데이터 예측을 통해서 최적의 재고관리 전략을 수립해봅시다.
- <https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting>

The screenshot shows the Kaggle competition page for 'Walmart Recruiting - Store Sales Forecasting'. The header is dark blue with a pattern of book icons. It includes the competition title, a subtitle 'Use historical markdown data to predict store sales', and metadata 'Walmart · 688 teams · 6 years ago'. Below the header is a navigation bar with links: Overview, Data, Notebooks, Discussion, Leaderboard, Rules, and a 'Join Competition' button. The main content area is titled 'Overview' and contains a 'Description' section. The description text reads: 'One challenge of modeling retail data is the need to make decisions based on limited history. If Christmas comes but once a year, so does the chance to see how strategic decisions impacted the bottom line.' Below the text are three promotional banners: 'CLEARANCE' with a red tag icon, 'Rollbacks' with a red arrow icon, and 'Special Buys' with a blue circular icon that says 'While SUPPLIES Last'.

Recruitment Prediction Competition

Walmart Recruiting - Store Sales Forecasting

Use historical markdown data to predict store sales

Walmart · 688 teams · 6 years ago

Overview Data Notebooks Discussion Leaderboard Rules [Join Competition](#)

Overview

Description

One challenge of modeling retail data is the need to make decisions based on limited history. If Christmas comes but once a year, so does the chance to see how strategic decisions impacted the bottom line.

Evaluation

Prizes

Timeline

CLEARANCE

Rollbacks

Special Buys

월마트 상품 판매량 판매량 수요예측 문제 정의

- **문제 정의** : 다른 지역에 있는 45개의 월마트 상점의 부서별 과거 판매량 데이터를 통해 **부서별 미래 판매량**을 예측해보자. (각 상점별로 여러개의 부서가 있습니다.)
- **판매량에 영향을 미치는 주요 특징** : 공휴일 기간에 시행하는 가격인하 프로모션 이벤트 (Markdown event)
- 판매량에 큰 영향을 미치는 **슈퍼볼(Super Bowl), 노동절(Labor Day), 추수감사절(Thanksgiving), 크리스마스(Christmas)** 4개의 주요 공휴일이 있고, 이 날들이 포함된 주는 성능 측정시 5배의 가중치를 갖습니다.

성능 평가 방법

- weighted mean absolute error (WMAE)를 이용해 성능을 평가합니다.

$$WMAE = \frac{1}{\sum w_i} \sum_{i=1}^n w_i |y_i - \hat{y}_i|$$

위 수식에서 각각의 기호는 다음을 뜻합니다.

- n 은 데이터의 개수를 나타냅니다.
- \hat{y}_i 는 모델에 의해 예측된 판매량을 나타냅니다.
- y_i 는 실제 판매량을 나타냅니다.
- w_i 는 가중치(weight)를 나타냅니다. 공휴일이 포함된 주(week)이면 $w = 5$, 그렇지 않으면 $w=1$ 입니다.

데이터 설명(Data Description)

- **stores.csv**

이 csv 파일은 익명화된 45개 상점의 크기(size)와 타입(type)을 나타냅니다.

- **train.csv**

이것은 2010년 2월 5일부터 2012년 11월 1일까지의 트레이닝 데이터를 나타냅니다. 컬럼은 다음과 같습니다.

- ① Store - 스토어를 나타내는 숫자
- ② Dept - 부서를 나타내는 숫자
- ③ Date - 날짜
- ④ Weekly_Sales - 부서별 주간 판매량
- ⑤ IsHoliday - 공휴일 포함 유무

데이터 설명(Data Description)

- **test.csv**

주간 판매량(Weekly_Sales)을 제외하고는 train.csv와 동일 형태의 파일입니다. 이 파일의 주간 판매량(스토어, 부서, 날짜별)을 예측하는 것이 Competition의 목적입니다.

- **features.csv**

이 csv 파일은 추가적인 특징정보를 포함하고 있습니다. 주요 컬럼들은 다음과 같습니다.

- ① Store - 스토어 숫자
- ② Date - 날짜
- ③ Temperature - 상점이 위치한 지역의 평균 온도
- ④ Fuel_Price - 상점이 위치한 지역의 기름값
- ⑤ Markdown1-5 - 익명화된 월마트의 프로모션 데이터. 모든 기간에 데이터가 존재하는 것은 아니며 결측치는 NA로 표기되어 있습니다.
- ⑥ CPI - 소비자 물가지수
- ⑦ Unemployment - 실업률
- ⑧ IsHoliday - 공휴일이 포함된 주인지 아닌지를 나타냄

데이터 설명(Data Description)

- 각 주요 공휴일에 대응되는 날짜는 다음과 같습니다.
- ① Super Bowl: 12-Feb-10, 11-Feb-11, 10-Feb-12, 8-Feb-13
- ② Labor Day: 10-Sep-10, 9-Sep-11, 7-Sep-12, 6-Sep-13
- ③ Thanksgiving: 26-Nov-10, 25-Nov-11, 23-Nov-12, 29-Nov-13
- ④ Christmas: 31-Dec-10, 30-Dec-11, 28-Dec-12, 27-Dec-13

실전 예제 2. 대형마트의 상품 판매량 예측 프로젝트 (월마트, 매장별로 내일 얼마나 팔릴까?)

< 5강_실전예제2_대형마트_상품판매량_예측하기.ipynb >

Step Up! 오늘 배운 내용 마스터하기

- 마케팅 분석 관련 추천 도서 – 린 분석(Lean Analytics)

<http://www.yes24.com/Product/Goods/11775117>

- scikit-learn Random Forest Regressor 설명

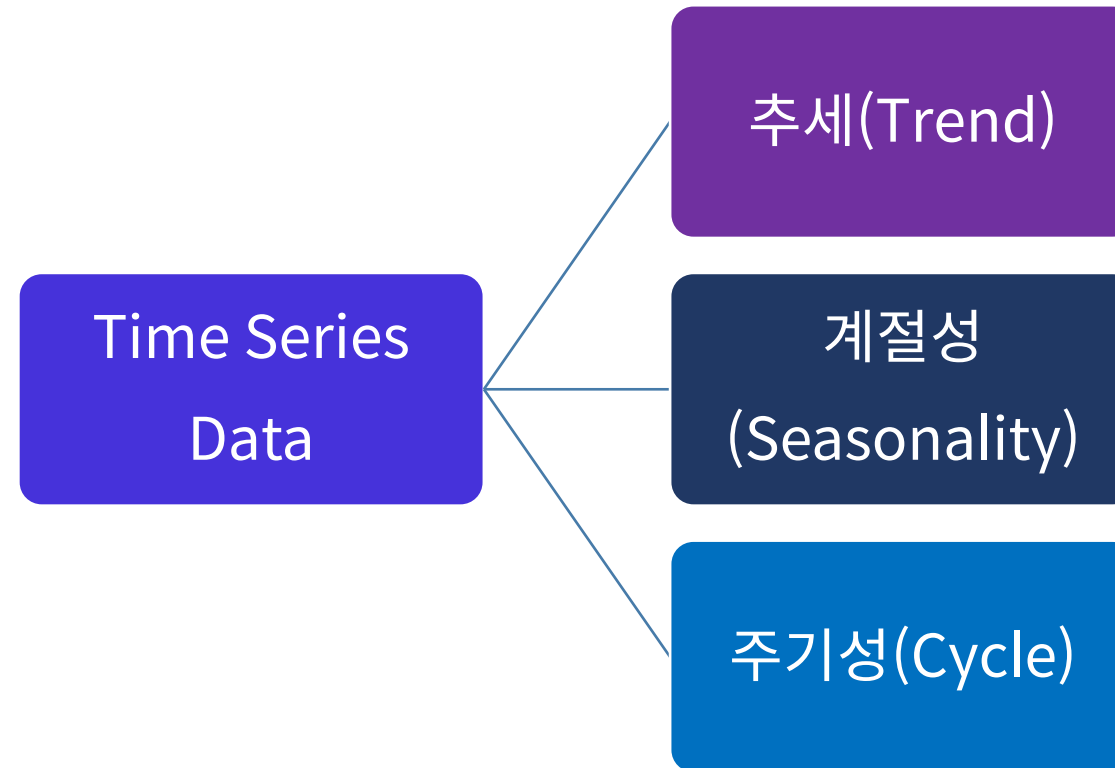
<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html>

- 코호트 분석 설명

<https://analyticsmarketing.co.kr/digital-analytics/google-analytics/1527/>

시계열 데이터의 구성 요소 – Trend, Seasonal, Cycle

- 시계열 데이터는 추세(Trend), 계절성(Seasonality), 주기성(Cycle)으로 분해될 수 있습니다.



Regression 알고리즘의 성능 평가 지표- RMSE(Root Mean Squared Error)

- MSE는 차이를 제공해서 더하므로 차이가 증폭되는 문제가 있을 수 있습니다.
- 따라서 MSE에 Root를 씌운 형태의 **RMSE (Root Mean Squared Error)**도 많이 사용하는 지표중 하나입니다.
- RMSE는 다음 수식으로 정의됩니다.

$$RMSE = \sqrt{\frac{1}{2n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}$$

Regression 알고리즘의 성능 평가 지표- MAE(Mean Absolute Error)

- 또한 예측값과 정답간의 차이에 절대값을 취한 **MAE(Mean Absolute Error)**도 Regression 알고리즘의 성능 평가 지표로 활용될 수 있습니다.
- **MAE**는 다음 수식으로 정의됩니다.

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|$$

팀프로젝트 Week2 – 자비스앤빌런즈

- 컬럼 정보는 다음과 같습니다.

- age: 고객의 만 나이입니다.
- gender: 고객의 성별입니다.
- year: 소득이 발생한 연도(귀속년도)입니다.
- refund: 예상환급액입니다.
- fee: 수수료입니다.
- has_paid: 수수료를 결제했는지의 여부
- income_근로: 고객의 근로소득(월급/일용직급여)
- income_사업: 고객의 사업소득(프리랜서 소득)
- income_기타: 고객의 기타소득(그외 기타 소득)

- 아래의 질문에 나름의 방식으로 답을 찾아주세요.

- (정답은 없습니다. 문제를 정의하고, 해답에 이르게 된 과정과 그 해석을 잘 설득/설명해 주시는 것이 중요합니다.)
- 고객의 결제여부에 영향을 미치는 요인들은 무엇인가요?
- 고객의 수수료 결제금액의 합을 높이기 위해서는 어떻게 해야 할까요?

팀프로젝트 진행방법

- 1) 팀장은 팀별로 main 저장소를 하나 생성합니다.
- 2) 각 팀원은 개별 작업내역은 main 저장소의 fork를 통해 작업합니다.
- 3) 분석 아이디어 및 협업은 main 저장소에 Issue를 통해 공유합니다.
- 4) 다음주 수업시작전에 팀원들간의 조율을 통해 작업내역을 main 저장소에 merge합니다.
- 5) 다음주에 팀별로 분석결과를 다른 사람들과 공유합니다.

Q & A



THANK YOU :)