DB HiTek

인공지능 활용 능력 개발 중급

[4차시]

AIBIZ
AIBIZ.Co.Ltd

프로젝트 목표

“인공지능 프로젝트 수행 순서 체득”

프로젝트 목표

- **인공지능 프로젝트 수행 단계**

단계	설명
목표 설정	- 프로젝트의 목표를 이해하고, 이를 데이터 수집 목표로 정의 - 프로젝트에 영향을 주는 중요한 항목 도출
데이터 이해	- 초기 데이터를 수집하고, 데이터의 품질 정의 - 가설을 위한 데이터 셋 정의
데이터 준비	- 분석 모델링에 필요한 데이터 추출 및 정제
모형	- 분석 기법을 선택하고, 분석에 필요한 최적 변수 설정 - 분석 모델 구축
평가	- 분석 모델에 대해 평가하고, 비즈니스 목표를 달성할 분석 모델 선정 - 전체 프로세스를 재검토하고, 다음 단계를 결정
적용	- 분석 모델링을 통해 획득한 지식 가공 - 보고서 작성 및 시각화

Scikit-learn 소개

- **Scikit-Learn 이란 ?**

I. python을 대표하는 머신러닝 라이브러리로 '사이킷런'이라고 부르기도함

II. 머신러닝 알고리즘 뿐만 아니라, 정규화, 데이터 분할 등 머신러닝 활용에 필요한 많은 모듈을 포함

Scikit-Learn Cheat Sheet

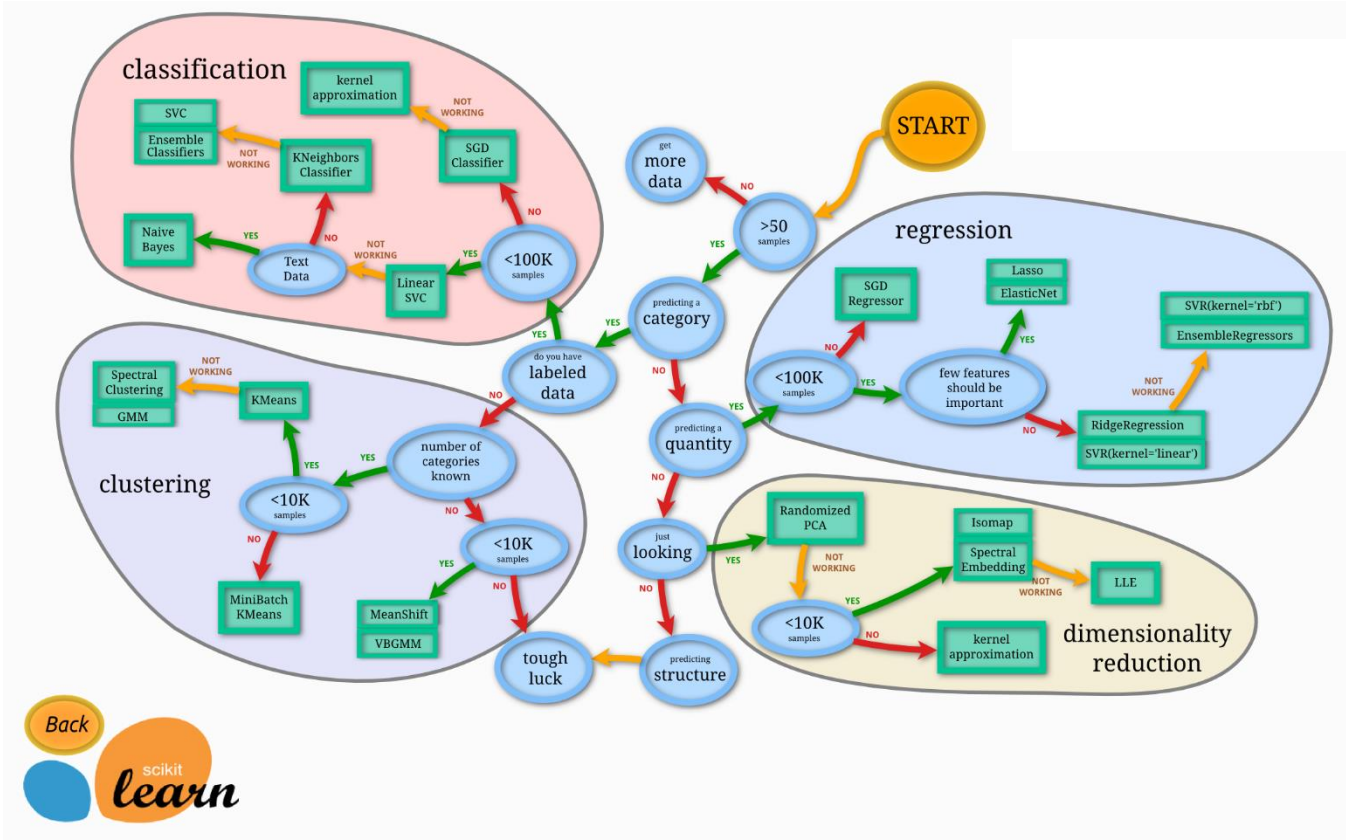


그림 출처 : sklearn-learn.org

Scikit-learn 소개

- **Scikit-Learn 사용법 ?**

I. 일반적으로 임포트 → 데이터로드 → 데이터분할 → 모델지정 → 모델학습 → 테스트 순으로 사용

```
# 1. Scikit-Learn 임포트
from sklearn.model_selection
import train_test_split
import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn
import datasets
# 2. 데이터 로드
iris = datasets.load_iris()
data, target = iris.data, iris.target
# 3. 데이터 분할(학습 데이터, 테스트 데이터)
X_train, X_test, y_train, y_test = train_test_split(data, target, random_state=0)
# 4. 모델 지정
model = RandomForestClassifier()
# 5. 모델 학습
model.fit(X_train, y_train)
# 6. 테스트 및 평가
y_pred = model.predict(X_test) print( '정답률 : ', accuracy_score(y_test, y_pred) )
```

프로젝트 목차

Project 1 : 반도체 공정데이터를 활용한 공정이상 예측 (분류)

Project 2 : 증착 공정 가상 계측 모델링 (회귀)

→ 제출물 : Project 1,2 Code

분류 Model 평가 지표

- **평가 지표 산출**

I. Confusion Matrix : Training을 통한 Prediction 성능을 측정하기 위해 예측 value와 실제 value를 비교하기 위한 표

		PREDICTED	
		Positive	Negative
ACTUAL	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

TP : Model이 1로 예측하고 정답도 1으로 정답인 경우
FP : Model이 1로 예측하고 정답이 0으로 오답인 경우
FN : Model이 0로 예측하고 정답이 1으로 오답인 경우
TN : Model이 0로 예측하고 정답도 0으로 정답인 경우

분류 Model 평가 지표

- **평가 지표 산출**

Confusion Matrix

ACTUAL	PREDICTED: Positive	PREDICTED: Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

지표 수식

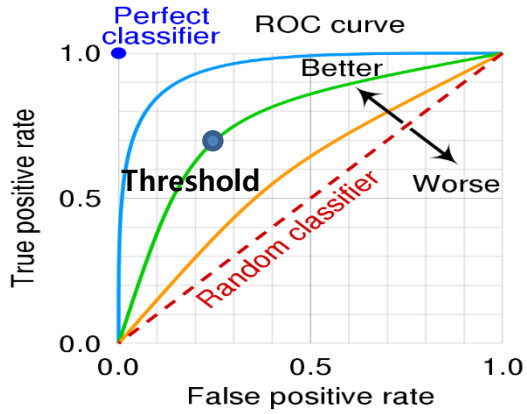
- Accuracy(정확도) = $\frac{TP + TN}{TP + TN + FP + FN}$
- Precision(정밀도) = $\frac{TP}{TP + FP}$
- Recall(재현도) = $\frac{TP}{TP + FN}$
- F-1 = $\frac{2*Recall*Precision}{Recall + Precision}$

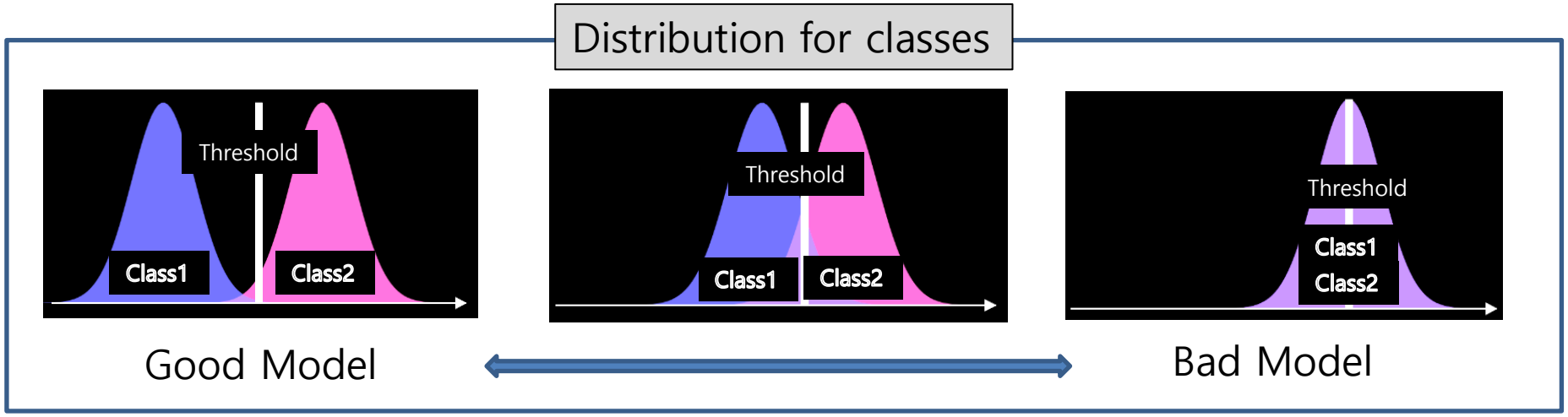
Accuracy : 올바르게 예측된 데이터의 수를 전체 데이터의 수로 나눈 값
→ 가장 직관적이나 데이터 불균형에 취약
Recall : 실제로 불량인 데이터를 모델이 불량이라고 인식한 데이터의 수
Precision : 모델이 불량으로 예측한 데이터 중 실제로 불량인 데이터의 수
→ Recall과 Precision은 Trade-Off 관계에 있으며, 적용 분야에 따라 중요도가 달라짐
F-1 Score : precision 과 recall의 조화평균

분류 Model 평가 지표

- **평가 지표 산출**

I. ROC Curve (Area Under Curve) : 모든 임계 값에서 분류 모델의 성능을 보여주는 그래프
→ Threshold를 결정하는 지표로 많이 사용





1. 반도체 공정데이터를 활용한 공정이상 예측

- **목표 설정**

  I. 반도체 공정 데이터 (센서데이터) 분석을 통하여 공정 이상을 예측

  II. 공정 이상에 영향을 미치는 요소들에 대한 데이터 분석

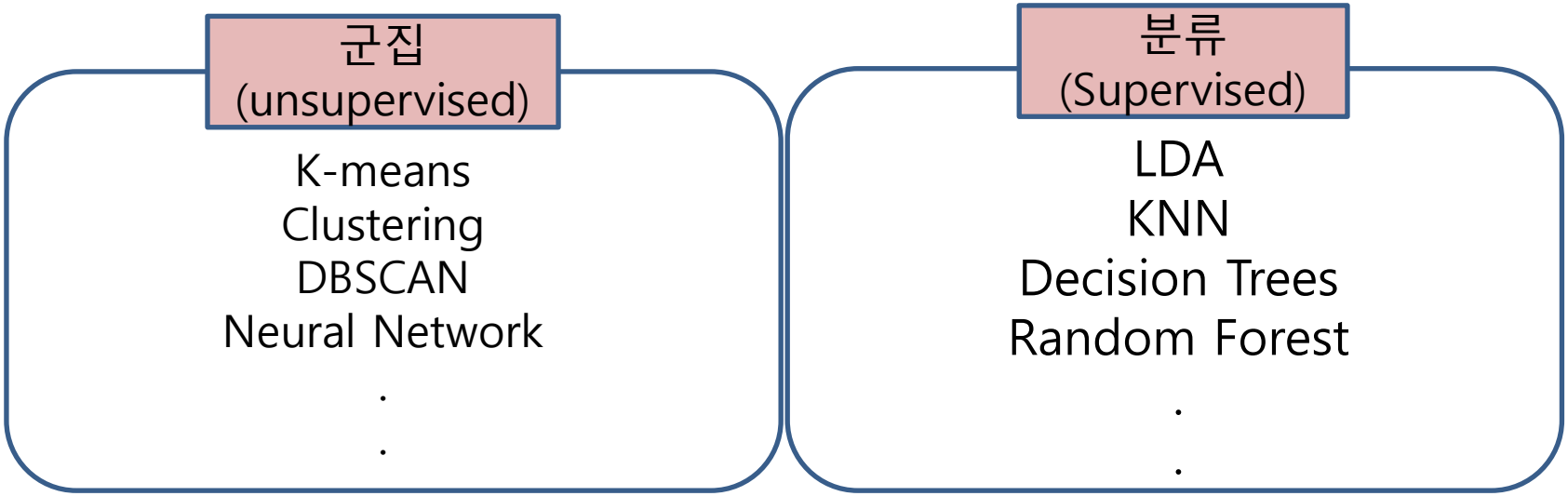
  → 분류, 군집 분석을 통해 공정 이상 예측을 하자

- **데이터의 이해 및 준비**

  I. "Labeling" 이 되어 있는 시계열 데이터 (센서 데이터)

  → Supervised Learning을 사용하자

- **모델**



1. 반도체 공정데이터를 활용한 공정이상 예측

- **평가**

Model	F-1	Recall	Precision	Accuracy
Random Forest	0.6	0.6	0.8	0.9
SVM	0.9	0.8	0.6	0.6

→ 정상 데이터가 1000개, 불량이 10개라면 어떤 모델을 선택 해야 할까요?

- **적용**

I. 보고서 작성, 시각화

1. 반도체 공정데이터를 활용한 공정이상 예측

- **프로젝트 목차**

1. **데이터 읽기:** 반도체 공정(SECOM) 데이터를 불러오고 Dataframe 구조를 확인

2. **데이터 정제:** 비어 있는 데이터 또는 쓸모 없는 데이터를 대체

3. **데이터 시각화:** 변수 시각화를 통하여 분포 파악
   3.1. Pass/Fail 시각화
   3.2. 센서 데이터 시각화 하기
   3.3. 59번 센서 데이터 시각화 하기

4. **데이터 전 처리:** 머신러닝 모델에 필요한 입력값 형식으로 데이터 처리
   4.1. x와 y로 분리
   4.2. 데이터 정규화

5. **머신러닝 모델 학습:** 분류 모델을 사용하여 학습 수행
   5.1. 기본 분류 모델 학습 - 로지스틱 분류기
   5.2. 다양한 분류 모델 학습

6. **평가 및 예측:** 학습된 모델을 바탕으로 평가 및 예측 수행
   6.1. Confusion Matrix
   6.2. Precision & Recall
   6.3. 테스트 데이터의 예측값 출력

1. 반도체 공정데이터를 활용한 공정이상 예측

- **데이터 정제**

1. 결측치 제거

I. Fixed Value : df.fillna()

Value : 결측값을 대체할 값입니다.
Method : 결측값을 변경할 방식입니다. { ' bfill',‘ffill ' }
- bfill로 할경우 결측값을 바로 아래 값과 동일하게 변경합니다.
- fill로 할 경우 결측값을 바로 위 값과 동일하게 변경합니다.

Original

DF	0	1	2	3
0	N/A	0.5	0.1	0.5
1	0.2	N/A	0.2	N/A
2	0.3	0.5	N/A	0.3
3	0.5	0.04	0.04	0.5

df.fillna(0)

DF	0	1	2	3
0	0	0.5	0.1	0.5
1	0.2	0	0.2	0
2	0.3	0.5	0	0.3
3	0.5	0.04	0.04	0.5

df.fillna(df.mean())

DF	0	1	2	3
0	0.33	0.5	0.1	0.5
1	0.2	0.34	0.2	0.43
2	0.3	0.5	0.11	0.3
3	0.5	0.04	0.04	0.5

df.fillna(methode=“bfill”)

DF	0	1	2	3
0	0.2	0.5	0.1	0.5
1	0.2	0.5	0.2	0.3
2	0.3	0.5	0.04	0.3
3	0.5	0.04	0.04	0.5

df.fillna(methode=“ffill”)

DF	0	1	2	3
0	N/A	0.5	0.1	0.5
1	0.2	0.5	0.2	0.5
2	0.3	0.5	0.2	0.3
3	0.5	0.04	0.04	0.5

1. 반도체 공정데이터를 활용한 공정이상 예측

- **데이터 정제**

1. 결측치 제거

I. Interpolation (보간법) : df.interpolate()

→ 보통 시계열 데이터를 활용할때 많이 사용됨

Method : 결측값을 변경할 방식입니다. {'linear','time'}
- linear로 할 경우, 결측값에 linear하게 증가,감소한 값으로 채워집니다.
- Time로 할 경우 결측값을 시간에 기준하여 증가,감소한 값으로 채워집니다.

Original

DF	Sensor #1
2016-12-01	1
2016-12-03	N/A
2016-12-04	N/A
2016-12-10	10

df.interpolate("linear")

DF	Sensor #1
2016-12-01	1
2016-12-03	4
2016-12-04	7
2016-12-10	10

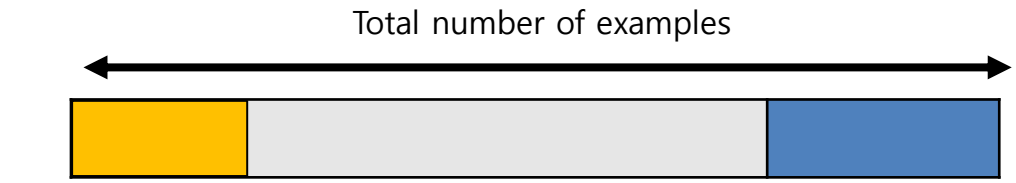
df.interpolate("linear")

DF	Sensor #1
2016-12-01	1
2016-12-03	3
2016-12-04	4
2016-12-10	10

1. 반도체 공정데이터를 활용한 공정이상 예측

- **데이터 분할**

I. 정제된 데이터를 머신러닝 알고리즘에 넣을 수 있도록 분할 (X : Feature, Y : Target)



Validation (x,y) Train (x,y) Test (x,y)

분할 순서

sklearn Import

```
from sklearn.cross_validation import train_test_split
Import pandas as pd
```

Target, Feature 분할

```
x = data.drop(columns = ['Target'], axis = 1)
y = data['Target']
```

Train, Test 데이터 추출

```
X_train, X_test, y_train, y_test
= train_test_split(X, y, test_size=0.2, random_state=1)
```

Validation 데이터 추출

```
X_train, X_val, y_train, y_val
= train_test_split(X_train, y_train, test_size=0.2, random_state=1)
```

*** Train시 K-Fold validation을 사용하면 자동으로 validation set 추출**

1. 반도체 공정데이터를 활용한 공정이상 예측

- **데이터 전처리**

1. 데이터 정규화, 표준화

I. Min-Max Normalization(정규화) :

→ 최소값을 빼고 범위로 나눔으로써 0-1 척도로 변환

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

II. Standardization(표준화) :

→ 평균을 빼고 표준 편차로 나눔

$$x' = \frac{x - \bar{x}}{\sigma}$$

Original

A	B	C
1000	10	0.5
765	5	0.35
800	7	0.09

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler

x = df.values
min_max_scaler = MinMaxScaler()
x_scaled = min_max_scaler.fit_transform(x)
df = pd.DataFrame(x_scaled)
```

A	B	C
1	1	1
0.765	0.5	0.7
0.8	0.7	0.18

Normalized

→ 데이터 정규화, 표준화중 어떤것을 선택 해야 할까요?

1. 반도체 공정데이터를 활용한 공정이상 예측

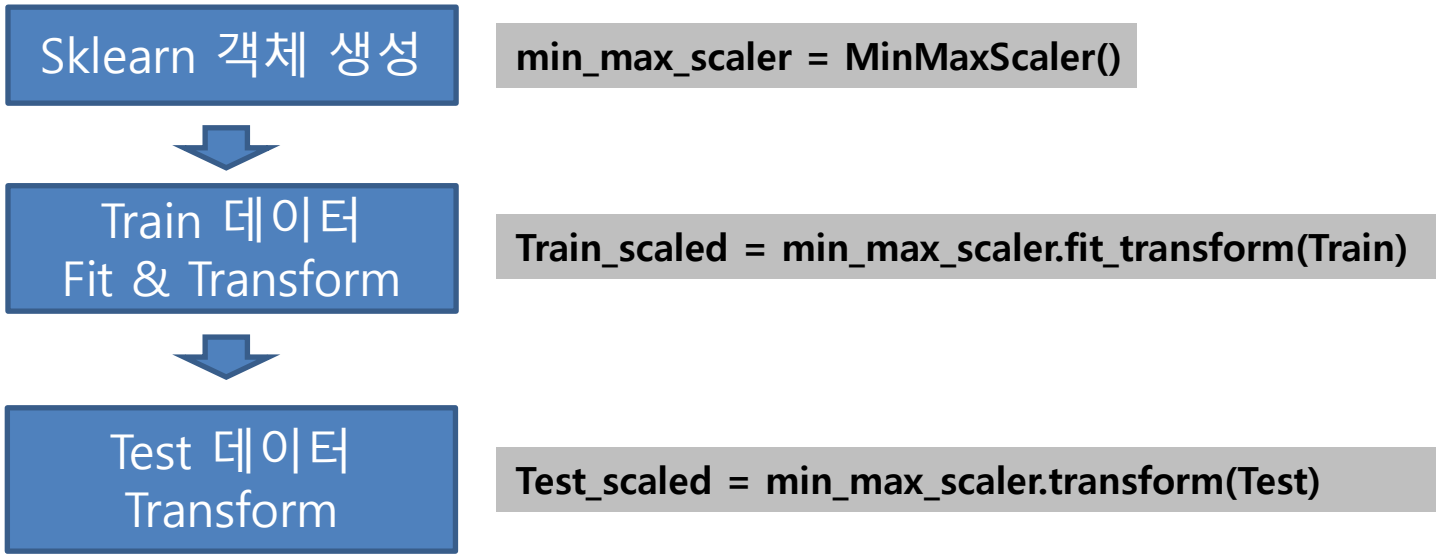
- **데이터 전처리**

1. 데이터 정규화, 표준화

→ Train 데이터에 Fit한 결과를 활용하여 Test 데이터에 적용 해야함

→ Normalization 이후, Train 데이터와 Test 데이터로 나누는 것도 좋은 방법

Normalization 순서



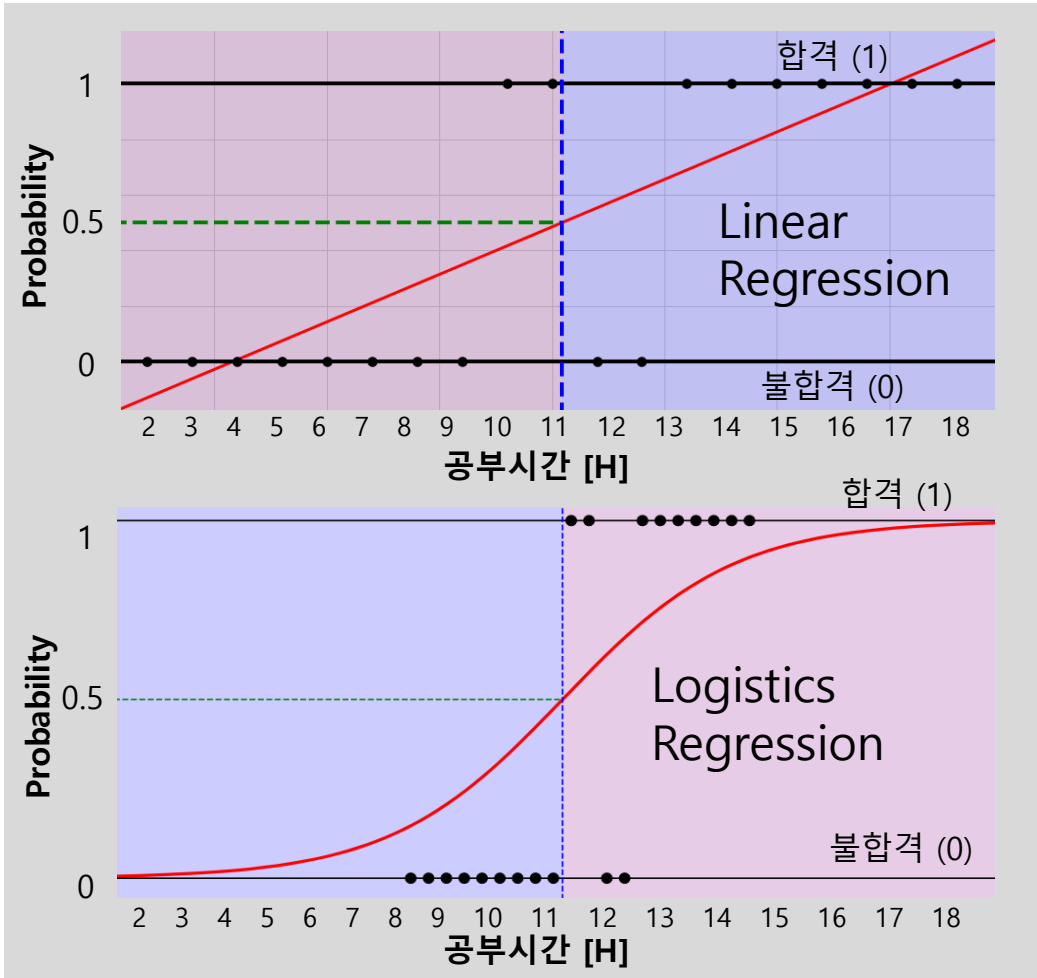
1. 반도체 공정데이터를 활용한 공정이상 예측

- **머신러닝 모델 학습**

1. 기본 분류 모델

I. 로지스틱 분류기 : 데이터를 0~1사이의 값을 갖는 Sigmoid 함수에 회귀하여 binary 분류

Features	Target
시간	Pss/Fail
1	1
2	0
3	1
4	1
.	.
.	.
.	.



1. 반도체 공정데이터를 활용한 공정이상 예측

- **평가**

Model	F-1	Recall	Precision	Accuracy
LDA				
KNN				
Decision Tree				
Random Forest				
SVM				

→ 어떤 모델을 선택 해야 할까요?

1. 반도체 공정데이터를 활용한 공정이상 예측

1. 모델 성능을 올리는 레시피

I. Data-Centric AI :

코드 & 알고리즘은 고정되어 있고, 데이터의 질만 반복적으로 향상 하는데 집중하는 방식

일관성 있는 데이터 레이블을 유지하는 것이 중요 (앤드류 응)

II. Model-Centric AI :

전형적인 전처리과정 후에 고정되어 있고 모델만 반복적으로 향상하는데 집중하는 방식

데이터를 최대한 모을 수 있는 만큼 모으고 데이터에 노이즈가 있더라도 문제 없을 정도로 모델을 최적화하는 것에 집중

	Steel defect Detection	Solar Panel	Surface Inspection
Baseline	76.2 %	75.68%	85.05%
Data-Centric	16.9%	+3.06%	*+0.4%*
Model-Centric	0%	+0.04%	*+0%*

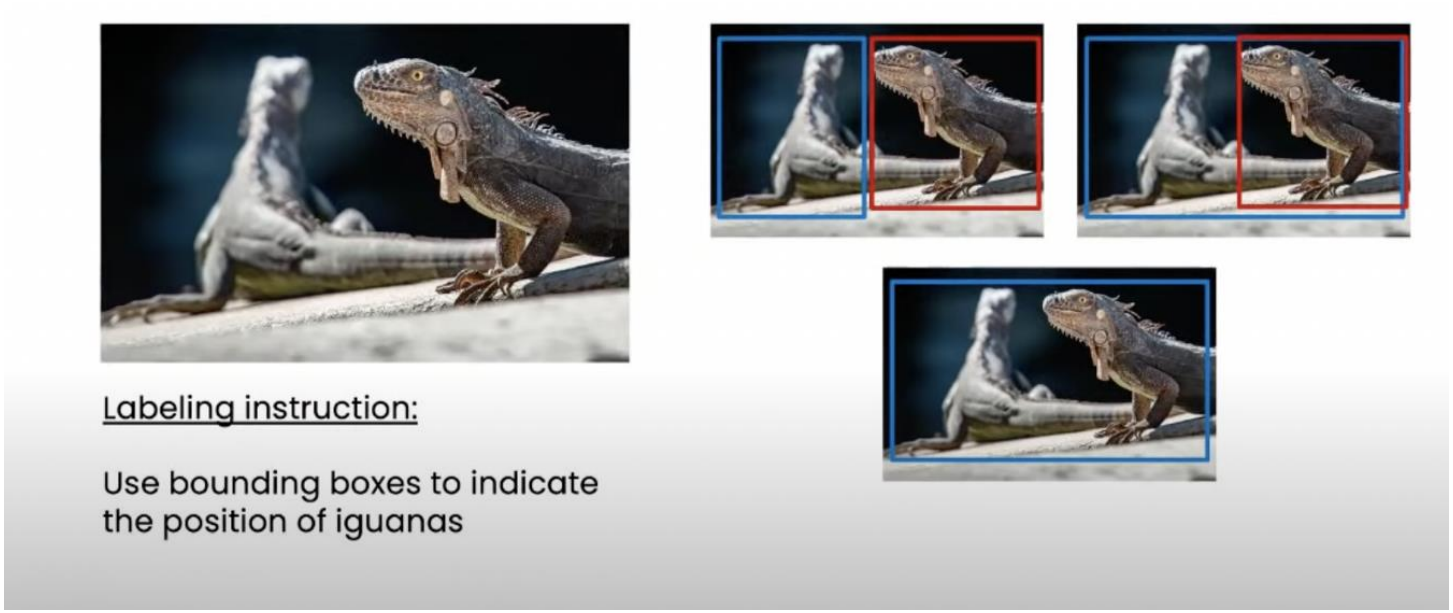
Improving code vs improving data quality 출처: Deeplearning.AI

1. 반도체 공정데이터를 활용한 공정이상 예측

1. 모델 성능을 올리는 레시피

I. Data-Centric AI : 좋은 데이터란?

데이터의 일관성 ! 적절한 크기 ! 중요 케이스를 포함!

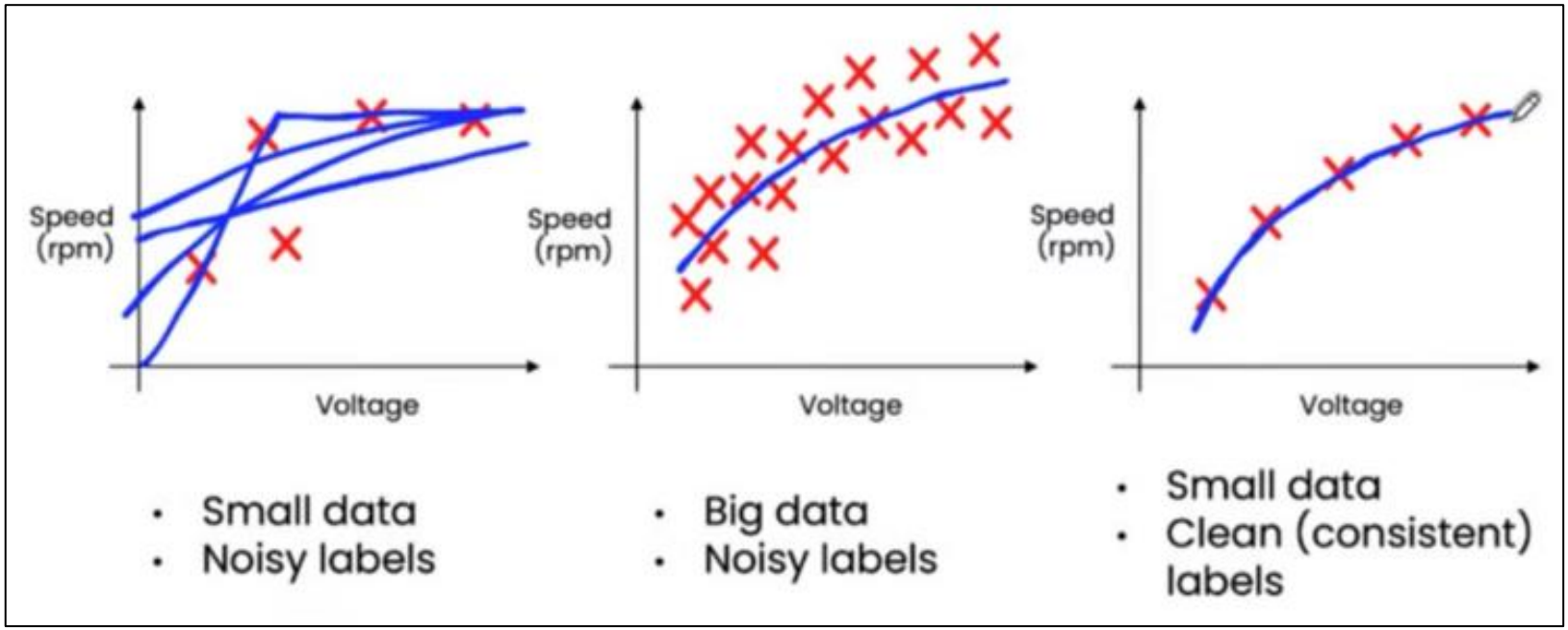


Improving code vs improving data quality 출처: Deeplearning.AI

1. 반도체 공정데이터를 활용한 공정이상 예측

1. 모델 성능을 올리는 레시피

I. Data-Centric AI : 데이터의 양과 질의 상관 관계

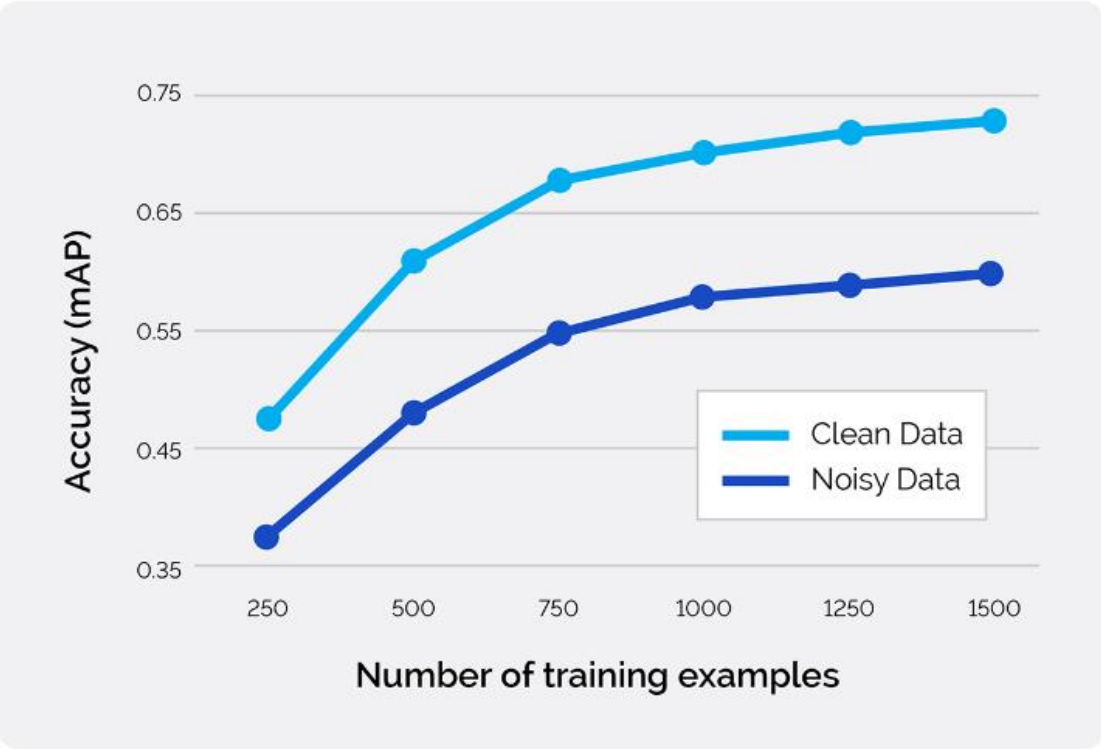


Improving code vs improving data quality 출처: Deeplearning.AI

1. 반도체 공정데이터를 활용한 공정이상 예측

1. 모델 성능을 올리는 레시피

I. Data-Centric AI : 데이터의 양과 질 중 무엇이 더 효율 적일까 ?



Improving code vs improving data quality 출처: Deeplearning.AI

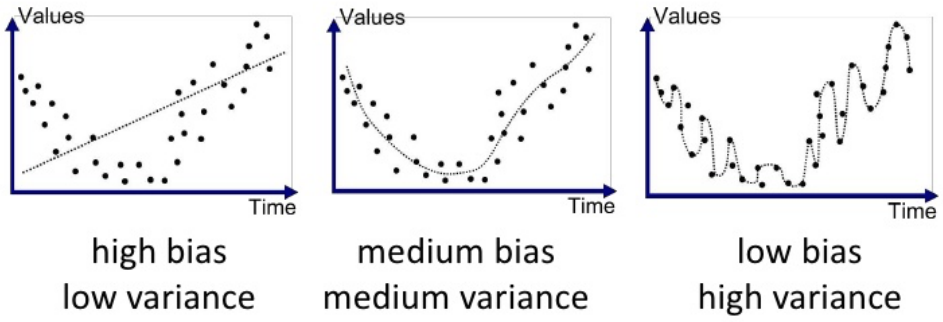
1. 반도체 공정데이터를 활용한 공정이상 예측

1. 모델 성능을 올리는 레시피

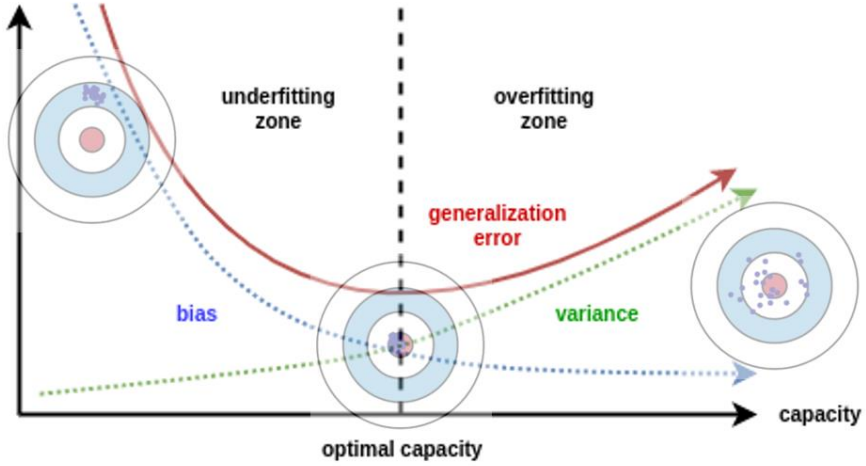
I. Model-Centric AI : 모델 성능을 올리기 위한 필수 개념

Bias : 모델을 통해 얻은 예측값과 실제 정답과의 차이의 평균

Variance : 다양한 데이터 셋에 대하여 예측 값이 얼만큼 변화할 수 있는 지에 대한 양(Quantity)의 개념



High bias 문제
Feature의 수를 늘림
좀 더 복잡한 모델을 사용
Regularization 줄임

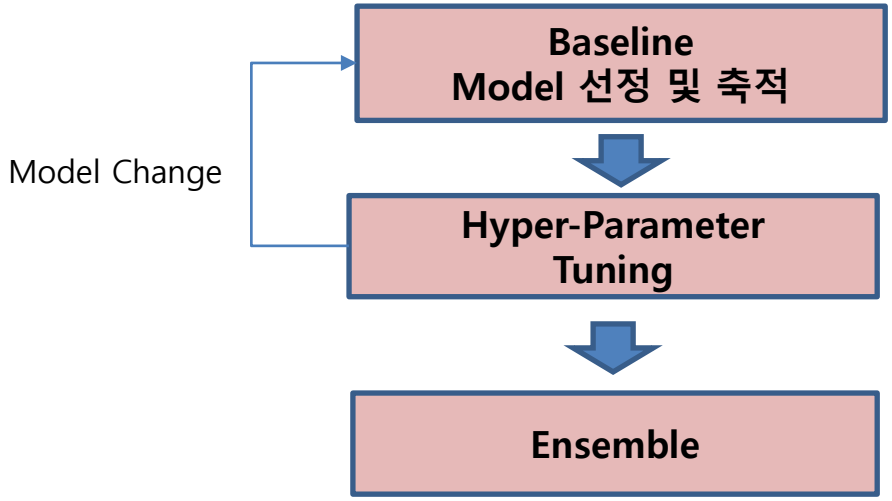


High variance 문제
training data의 수를 늘림
Feature의 수를 줄임
Regularization 강화

1. 반도체 공정데이터를 활용한 공정이상 예측

1. 모델 성능을 올리는 레시피

I. Model-Centric AI :

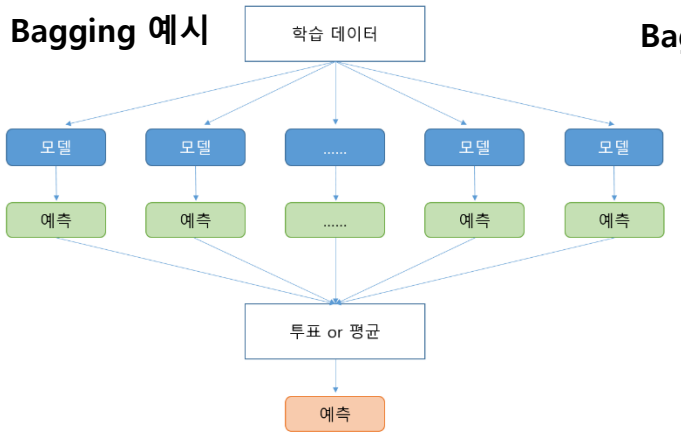


Machin Learning 알고리즘 평가
State-of-the-art (SOTA) 브라우저 검색

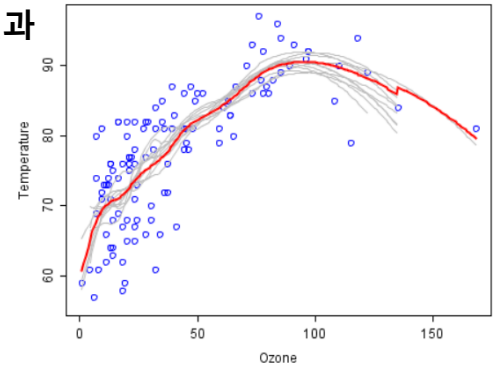
Learning Rate etc ...

Bootstrapping, Data Bagging,
Model Bagging

Bagging 예시



Bagging 결과



출처 : Wikipedia/Bootstrap_aggregating

프로젝트 목차

Project 1 : 반도체 공정데이터를 활용한 공정이상 예측 (분류)

Project 2 : 증착 공정 가상 계측 모델링 (회귀)

Project 3 : 고급 과정 Test

Keras 소개

- **Keras 사용법 ?**

I. 일반적으로 임포트 → 데이터로드 → 데이터분할 → 모델지정 → 모델학습 → 테스트 순으로 사용

```
# 1. Keras 임포트
import numpy as np
from tensorflow import keras
from tensorflow.keras import layers
# 2. 데이터 로드 및 분할
(x_train, y_train), (x_test, y_test) = keras.datasets.mnist.load_data()

# 3. 모델 지정
model = keras.Sequential(
    [   keras.Input(shape=input_shape)
        layers.Dropout(0.5),
        layers.Dense(num_classes, activation="softmax")])
# 4. 모델 학습
model.compile(loss="categorical_crossentropy", optimizer="adam", metrics=["accuracy"])
model.fit(x_train, y_train, batch_size=batch_size, epochs=epochs, validation_split=0.1)
# 5. 모델 학습
model.fit(X_train, y_train)
# 6. 테스트 및 평가
score = model.evaluate(x_test, y_test, verbose=0)
```

프로젝트 목표

- **인공지능 프로젝트 수행 단계**

단계	설명
목표 설정	- 프로젝트의 목표를 이해하고, 이를 데이터 수집 목표로 정의 - 프로젝트에 영향을 주는 중요한 항목 도출
데이터 이해	- 초기 데이터를 수집하고, 데이터의 품질 정의 - 가설을 위한 데이터 셋 정의
데이터 준비	- 분석 모델링에 필요한 데이터 추출 및 정제
모형	- 분석 기법을 선택하고, 분석에 필요한 최적 변수 설정 - 분석 모델 구축
평가	- 분석 모델에 대해 평가하고, 비즈니스 목표를 달성할 분석 모델 선정 - 전체 프로세스를 재검토하고, 다음 단계를 결정
적용	- 분석 모델링을 통해 획득한 지식 가공 - 보고서 작성 및 시각화

회귀 Model 평가 지표

- **평가 지표 산출**

I. Mean Square Error (MSE)
II. Root Mean Square Error (RMSE)
III. Mean Absolute Error (MAE)

$$MSE = \frac{1}{N}\sum_{i}^{N}(pred_i - target_i)^2$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i}^{N}(pred_i - target_i)^2}$$

$$MAE = \frac{1}{N}\sum_{i}^{N}|(pred_i - target_i)|$$

Epoch에 따른 예측 결과

Epoch	Prediction	Target
1	[0, 4, 9]	[3, 5 ,7]
2	[2, 4, 2]	[3, 5, 7]
3	[3, 5 ,6]	[3, 5, 7]

Error Score 산출 결과

Epoch	MSE	RMSE	MAE
1	$\frac{14}{3} \approx 4.6$	$\frac{\sqrt{14}}{3} \approx 1.2$	$\frac{6}{3} \approx 2$
2	$\frac{27}{3} \approx 9$	$\frac{\sqrt{27}}{3} \approx 1.7$	$\frac{7}{3} \approx 2.3$
3	$\frac{1}{3} \approx 0.33$	$\frac{\sqrt{3}}{3} \approx 0.57$	$\frac{1}{3} \approx 0.33$

회귀 Model 평가 지표

- **평가 지표 산출**

I. Mean Square Error (MSE)
→ 이상치에 약함

II. Root Mean Square Error (RMSE)
→ 이상치에 강건

III. Mean Absolute Error (MAE)
→ 이상치에 강건

Error는 작을수록 좋은 Model !

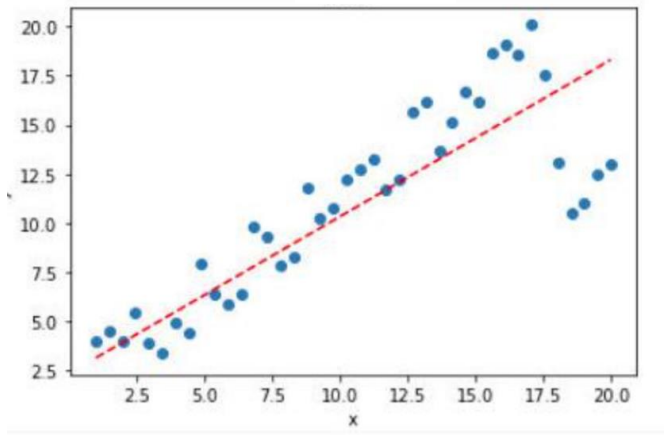
$$MSE = \frac{1}{N}\sum_{i}^{N}(pred_i - target_i)^2$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i}^{N}(pred_i - target_i)^2}$$

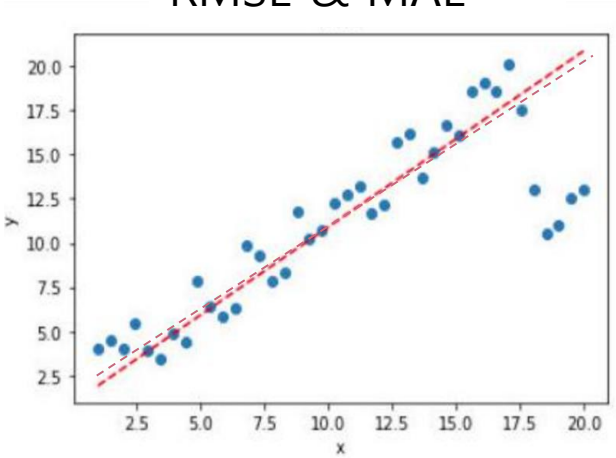
$$MAE = \frac{1}{N}\sum_{i}^{N}|(pred_i - target_i)|$$

Linear Regression 결과

MSE



RMSE & MAE



회귀 Model 평가 지표

- **평가 지표 산출**

I. Mean Square Error (MSE)
→ 모든 점에서 미분 가능
→ 큰 Error에 penalty를 줌

II. Root Mean Square Error (RMSE)
→ 미분 불가 점이 존재
→ 큰 Error에 penalty를 줌

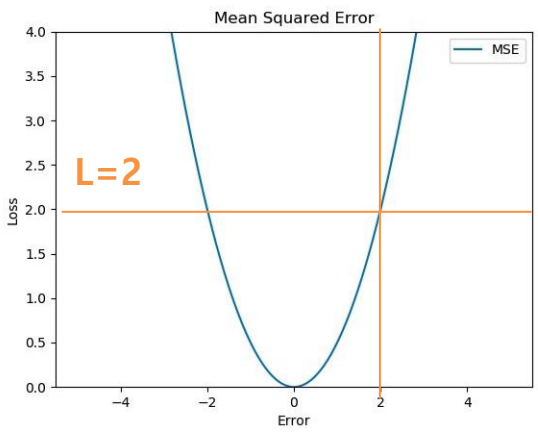
III. Mean Absolute Error (MAE)
→ 미분 불가 점이 존재
→ 모든 점에서 Error가 동일

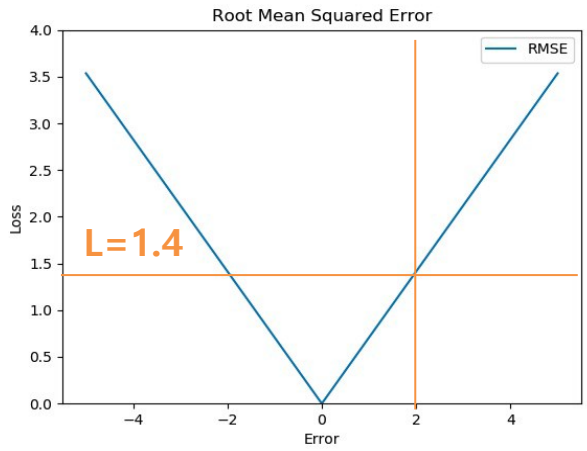
$$MSE = \frac{1}{N}\sum_{i}^{N}(pred_i - target_i)^2$$

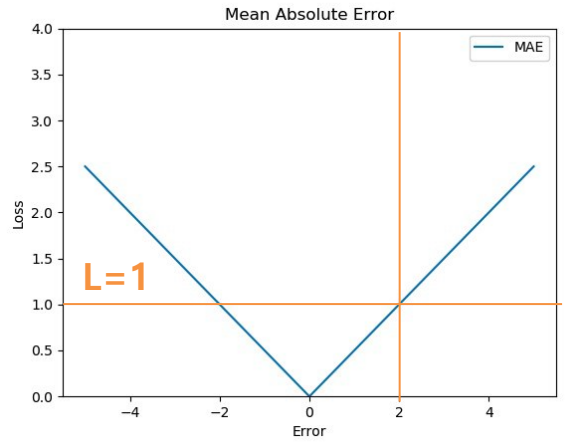
$$RMSE = \sqrt{\frac{1}{N}\sum_{i}^{N}(pred_i - target_i)^2}$$

$$MAE = \frac{1}{N}\sum_{i}^{N}|(pred_i - target_i)|$$

Error vs Loss 예시







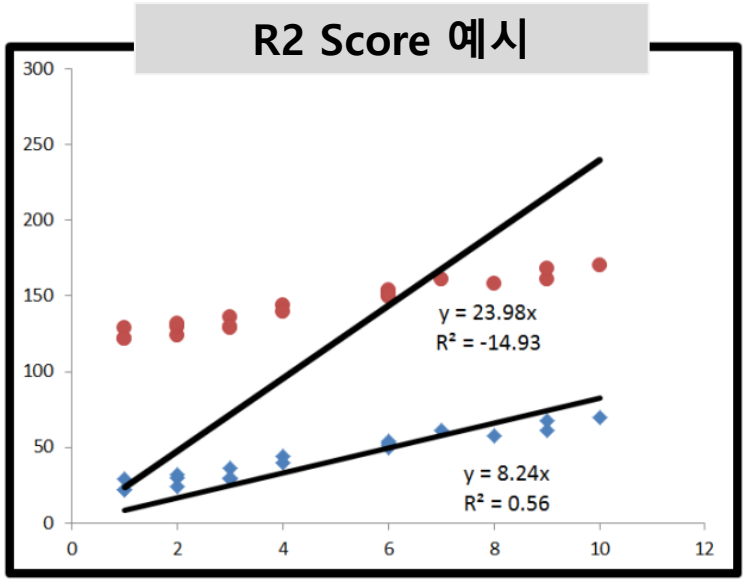
회귀 Model 평가 지표

- **평가 지표 산출**

I. R2 Score (R-squred)
→ Predict결과와 Ground Truth 결과를 직관적으로 비교 하도록 함
→ Excel에서도 많이 사용 하고 있음

$$R^2 = \frac{SSR}{SST} = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2}$$

where
SSR : Sum of squared regression
SST : Sum of squares total
$\bar{y}$ = Mean
$\hat{y}$ = Prediction
y = Ground Truth



2. 증착 공정 가상 계측 모델링

- **목표 설정**

  I. 공정 데이터를 활용한 박막 두께를 예측

  II. 증착 두께를 예측에 영향을 미치는 요소들에 대한 데이터 분석 및 시각화

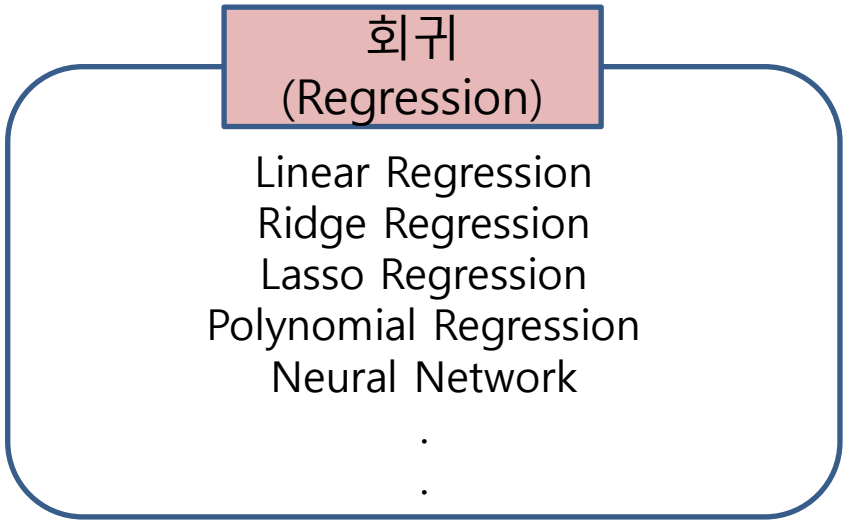
  → Regression model을 사용 하자

- **데이터의 이해 및 준비**

  I. "Labeling" 이 되어 있는 센서 데이터

  → Supervised Learning을 사용하자

- **모델**



2. 증착 공정 가상 계측 모델링

- **평가**

Model	MSE	RMSE	MAE	R-2
ANN				

- **적용**

I. 보고서 작성, 시각화

2. 증착 공정 가상 계측 모델링

- **프로젝트 목차**

1. **데이터 읽기 :** 데이터를 불러오고 Dataframe 구조를 확인

2. **데이터 시각화 :** 데이터를 시각화 하여 Outlier 확인

3. **데이터 정제 :**
   3.1. 결측치 제거
   3.2. 이상치 제거 (z-score 활용)

4. **데이터 전처리:** 머신러닝 모델에 필요한 입력값 형식으로 데이터 처리
   4.1. x와 y로 분리
   4.2. 데이터 정규화
   4.3. 학습데이터, 평가데이터 분리

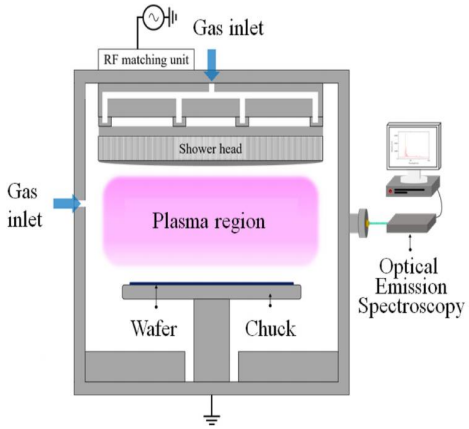
5. **딥러닝 모델 학습:** ANN 인공신경망을 활용한 회귀
   5.1. 인공신경망 모델 생성
   5.2. 인공신경망 모델 학습
   5.3. 인공신경망 학습 결과 시각화
   5.4. 인공신경망 학습 결과 평가

6. **평가 및 예측:** 학습된 모델을 바탕으로 평가 및 예측 수행
   6.1. 인공신경망 예측 결과 복원
   6.2. 인공신경망 가상 계측 및 시각화

2. 증착 공정 가상 계측 모델링

- **가상 계측 데이터 프리뷰**

1. **PECVD 장비의 RF power, pressure, C_(3) H_(6), N_(2) 변화 DOE 실험 데이터**



- 13.5 MHz RF powered CCP type PECVD
- 4요인 3수준 Box-Behnken DOE 27 run
- 두께 측정 : Reflectometer 사용

Input parameters	Ranges	Unit
RF Power	230-270	wait
Pressure	800-1200	mTorr
$C_3 H_6$	60-100	Sccm
N_2	30-50	Sccm

Output parameter	Ranges	Unit
Thickness	701 - 1801	A

2. 증착 공정 가상 계측 모델링

- **Outlier 제거**

1. Z-Score (표준 점수)

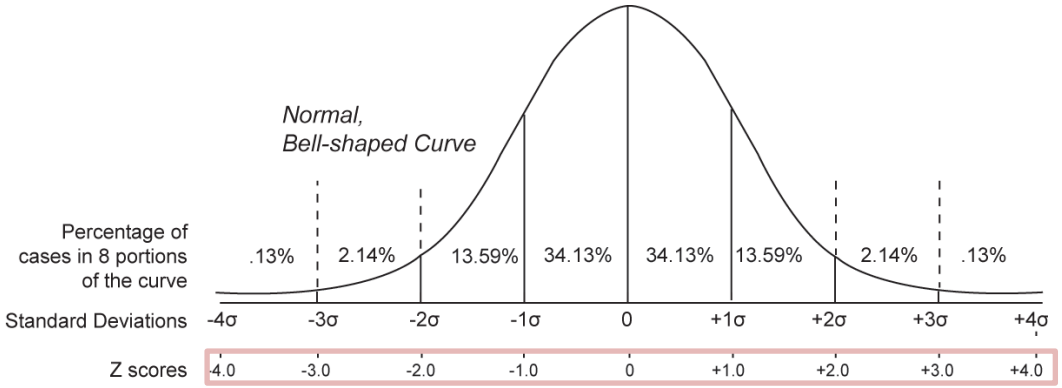
I. Normal distribution에서의 평균값과 얼만큼 떨어져 있는지 알려주는 수학적 지표

Z-Score 수식

$$Z = \frac{x - \mu}{\sigma}$$

Score, Mean, SD

Normal Distribution

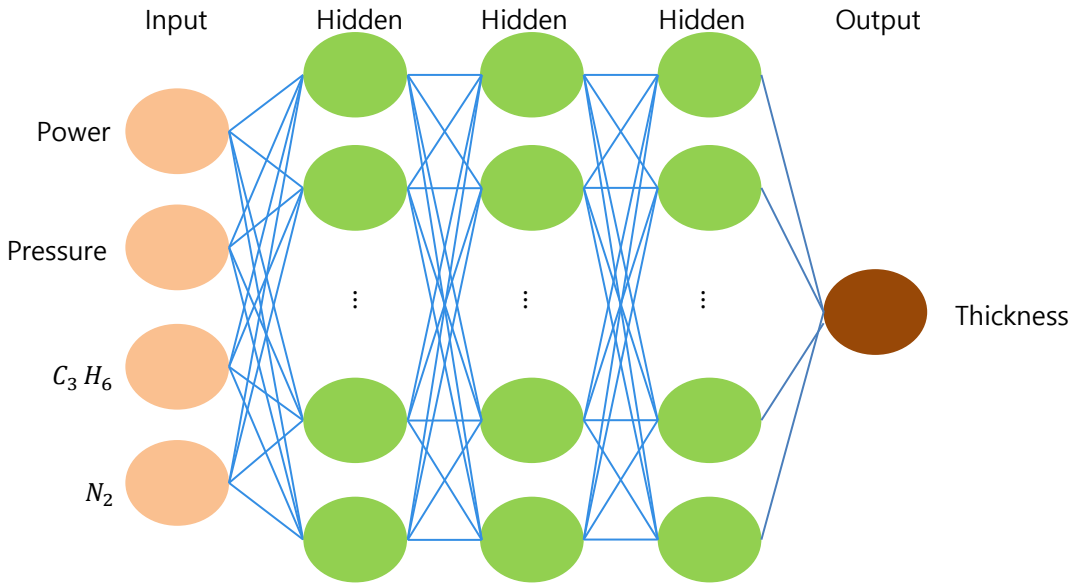


2. 증착 공정 가상 계측 모델링

- **Model 구성도**

1. Sequential Model

I. 순차적으로 layer를 쌓은 모델



- Input layer 1개, Output layer 1개, hidden layer 3개로 구성된 artificial neural network (ANN, 인공신경망) 구조

```
model = Sequential()

model.add(Dense(16, kernel_initializer = 'he_uniform', input_dim=4))
model.add(Activation('relu'))
model.add(Dense(32, kernel_initializer = 'he_uniform'))
model.add(Activation('relu'))
model.add(Dense(16, kernel_initializer = 'he_uniform'))
model.add(Activation('relu'))
model.add(Dense(1, activation='relu'))

adam = Adam(lr = 0.001)
model.compile(loss = 'mse', optimizer = adam, metrics = ['mae'])

model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_15 (Dense)            (None, 16)                80

 activation_12 (Activation)  (None, 16)                0

 dense_16 (Dense)            (None, 32)                544

 activation_13 (Activation)  (None, 32)                0

 dense_17 (Dense)            (None, 16)                528

 activation_14 (Activation)  (None, 16)                0

 dense_18 (Dense)            (None, 1)                 17

=================================================================
Total params: 1,169
Trainable params: 1,169
Non-trainable params: 0
_________________________________________________________________
```

2. 증착 공정 가상 계측 모델링

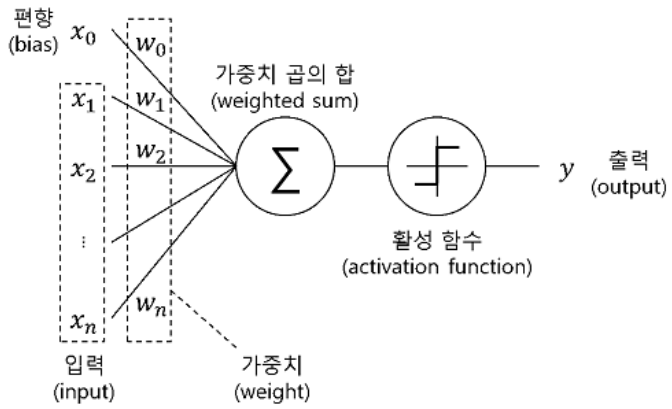
- **Model 구성도**

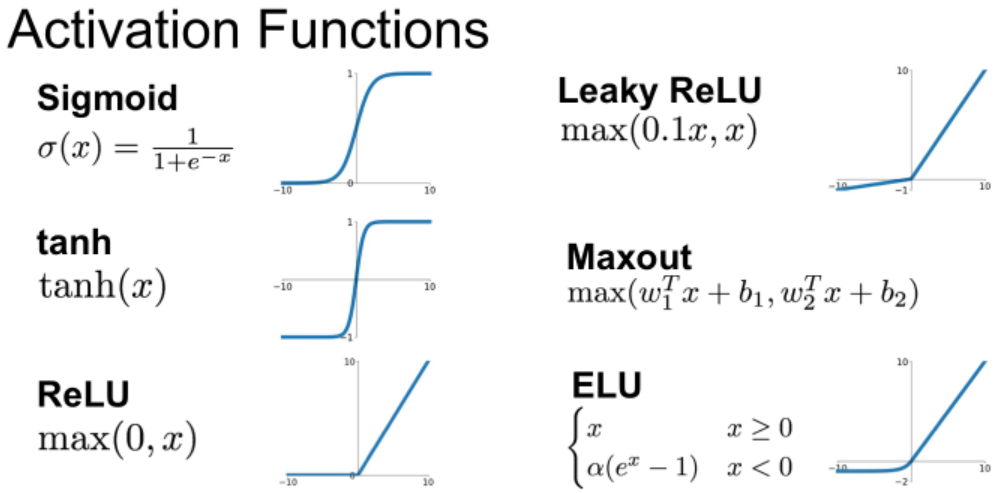
1. Activation Layer

I. 활성화 함수가 없으면 어떻게 될까?

이전 층(layer)의 결과값을 변환하여 다른 층의 뉴런으로 신호를 전달하는 역할을 하며,

활성화 함수가 필요한 이유는 활성화 함수가 없다면, linear한 문제만 해결이 가능함. 비선형 문제를 풀기 위해서는 활성화 함수가 반듯이 필요함





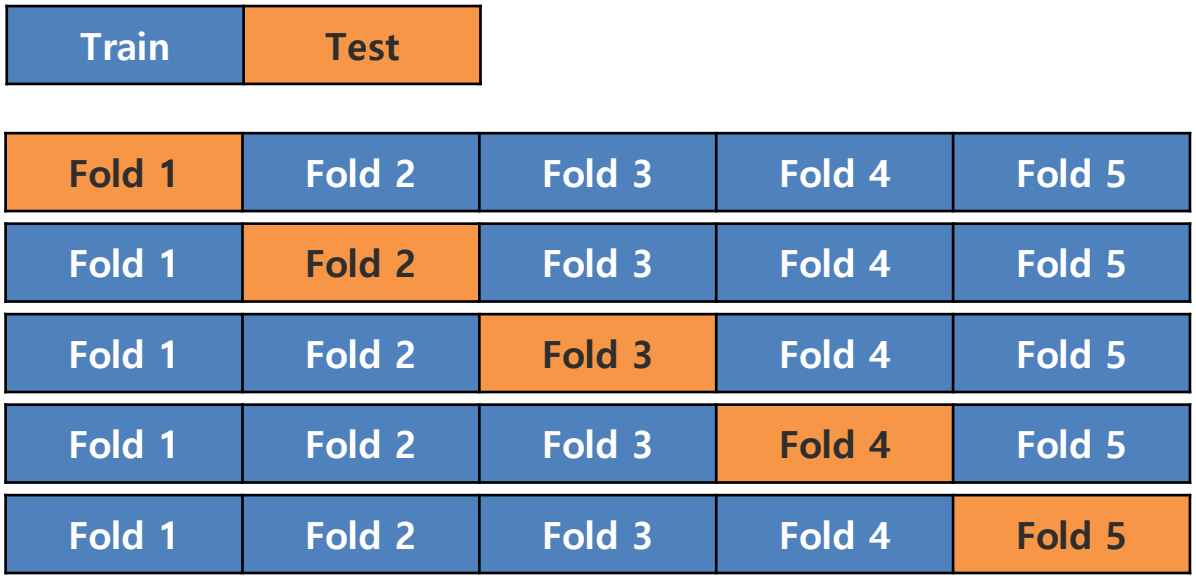
2. 증착 공정 가상 계측 모델링

- **학습 방법**

1. 학습 (Train) 데이터, 평가 (Test) 데이터 분리

I. Hold-out 검증
: Train, test의 비율을 나누어 검증 => 일반적으로 train, validation, test 3구간 분리

II. K-fold cross validation
: K의 개수대로 분리 후, 하나의 fold를 test에 사용하고 나머지를 train에 사용하는 방법
ex) k=5일 때, 5번 모델 학습을 반복하여 성능 지표의 평균을 평가



2. 증착 공정 가상 계측 모델링

- **학습 방법**

1. 학습 (Train) 데이터, 평가 (Test) 데이터 분리

I. Hold-out 검증
: Train, test의 비율을 나누어 검증 => 일반적으로 train, validation, test 3구간 분리


```
x_train, x_test, y_train, y_test = train_test_split(x_s, y_s, test_size=0.2, shuffle=True, random_state=100)
y_test
```

```
9      1006.362727
23     1404.362727
21     1639.362727
12     1800.362727
5      1000.362727
11     1283.362727
Name: Thickness, dtype: float64
```

- test_size = 0.2
  : 전체 데이터에서 test data의 개수를 20%로 분리
- Shuffle = True
  : train test split 시에 데이터를 섞어서 분리
- Random_state
  : 숫자 입력하여 random state 고정

2. 증착 공정 가상 계측 모델링

- **모델 평가**

1. MSE, RMSE, MAE 산출

I. Hold-out 검증
: Train, test의 비율을 나누어 검증 => 일반적으로 train, validation, test 3구간 분리

```
print("Mean squared error (MSE)_Train data :",round(mean_squared_error(y_train_i,y_train_pred_i),2))
print("Root mean squared error (RMSE)_Train data :",round(mean_squared_error(y_train_i,y_train_pred_i)**0.5,2))
print("Mean absolute error (MSE)_Train data :",round(mean_absolute_error(y_train_i,y_train_pred_i),2))
print("R2_Train data :",round(r2_score(y_train_i,y_train_pred_i),3))
```

```
Mean squared error (MSE)_Train data : 1718.94
Root mean squared error (RMSE)_Train data : 41.46
Mean absolute error (MSE)_Train data : 32.96
R2_Train data : 0.97
```

```
print("Mean squared error (MSE)_Test data :",round(mean_squared_error(y_test_i,y_test_pred_i),2))
print("Root mean squared error (RMSE)_Test data :",round(mean_squared_error(y_test_i,y_test_pred_i)**0.5,2))
print("Mean absolute error (MSE)_Test data :",round(mean_absolute_error(y_test_i,y_test_pred_i),2))
print("R2_Test data :",round(r2_score(y_test_i,y_test_pred_i),3))
```

```
Mean squared error (MSE)_Test data : 21000.46
Root mean squared error (RMSE)_Test data : 144.92
Mean absolute error (MSE)_Test data : 133.48
R2_Test data : 0.798
```

Train 정확도는 높지만 Test 정확도는 낮은 상태
=> 과적합 (Overfitting) 상태

2. 증착 공정 가상 계측 모델링

- **언더/오버 Fitting을 막는 Recipe**

I. 데이터의 수를 늘리자

II. Parameter 수를 줄이거나 늘리자 (Model Complexity)

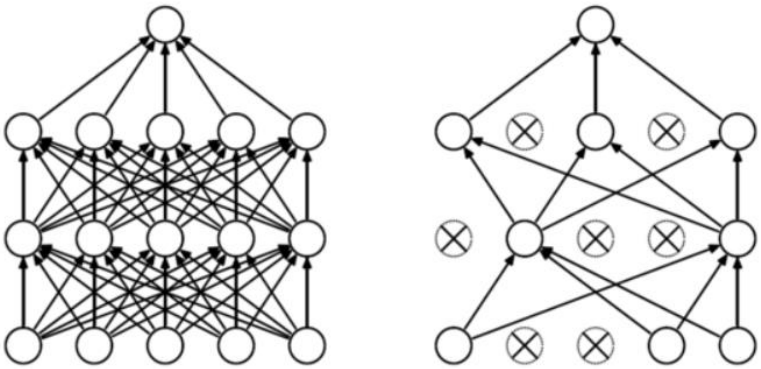
III. K-Fold Validation

IV. Regularization

V. Dropout layer : 네트워크의 유닛의 일부만 동작하고 일부는 동작하지 않도록 하는 방법

Activation layer 직후 사용하면 효과적

Dropout Layer



(a) Standard Neural Net (b) After applying dropout.

2. 증착 공정 가상 계측 모델링

Project 1 : 반도체 공정데이터를 활용한 공정이상 예측 (분류)

Project 2 : 증착 공정 가상 계측 모델링 (회귀)

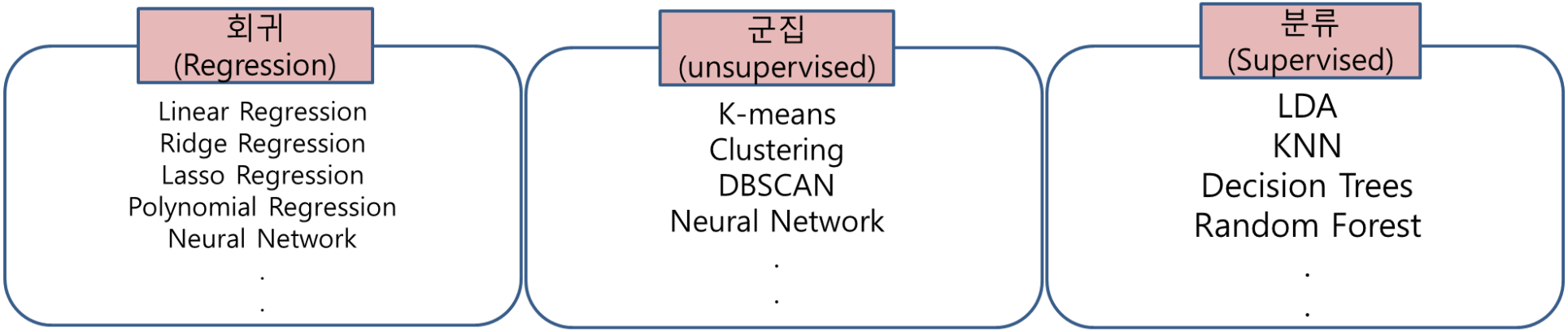
Project 3 : 고급 과정 Test

3. 고급 과정 Test

- **목표 설정**
  I. 목표 ?

- **데이터의 이해 및 준비**
  I. "Labeling" 여부 ?
  II. 데이터의 Shape ?
  III. 데이터의 속성 ?

- **모델**



3. 고급 과정 Test

- **평가**

Model	F-1	Recall	Precision	Accuracy
Random Forest				
SVM				

Model	MSE	RMSE	MAE	R-2
ANN				

3. 고급 과정 Test

- **적용**

I. 시각화 : Epoch에 따른 loss etc..

II. Hyper parameter 값 정리 : learning rate, epoch, etc..

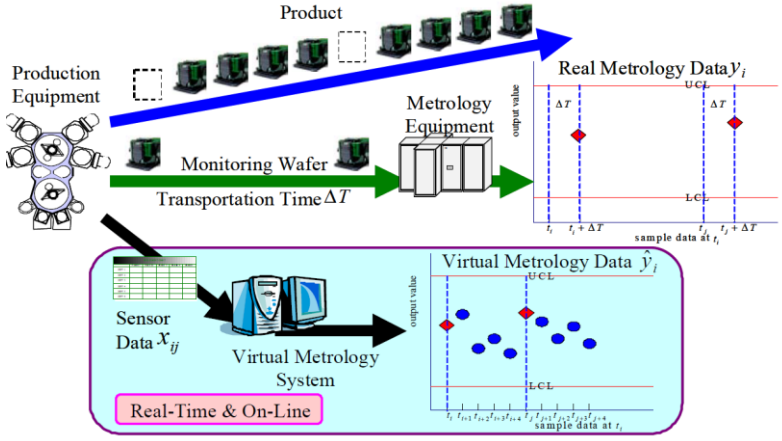
III. 성능 향상 방법 정리 : 추가적으로 어떤 Action을 하면 성능을 올릴 수 있을까 ?

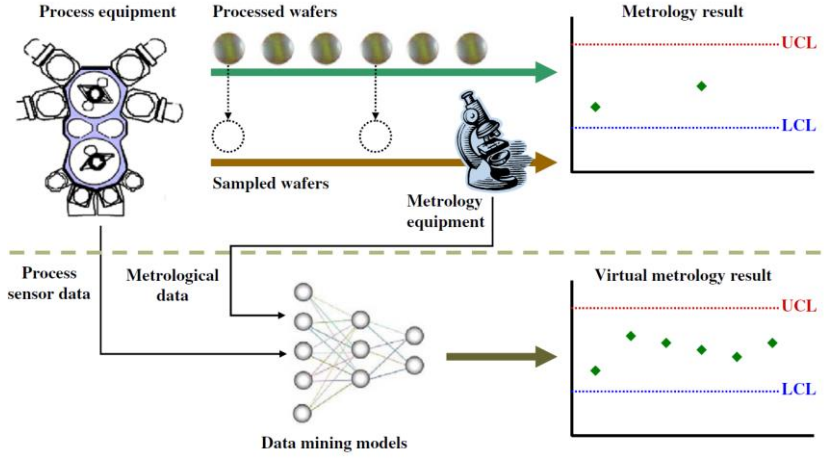
Appendix

2. 증착 공정 가상 계측 모델링 논문 리뷰

1. **논문 제목 : Machine learning-based virtual metrology on film thickness in ACL deposition process (ACL 증착 공정의 박막 두께를 예측하는 머신러닝 기반 가상 계측 연구)**
2. **Introduction**

- Moore's law => Critical dimension (CD) ↓, Manufacturing complexity ↑





- Metrology (계측)
  - 다음 공정을 위해 필수적인 과정
  - 모든 웨이퍼를 검사하려면 많은 계측 장비들이 필요
    -> Cost와 production cycle time 증가

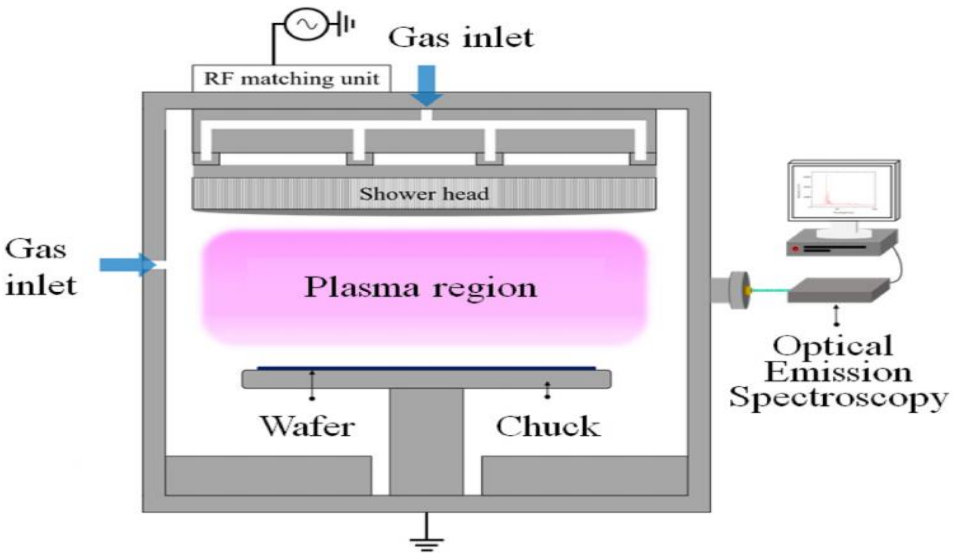
- 가상 계측(Virtual metrology, VM)의 장점
  - 실제 웨이퍼 측정에 요구되는 비용과 시간 단축
  - 웨이퍼 스크랩 감소
  - Metrology와 inspection 과정에서 throughput 증가

→ 복잡해진 공정 과정에서 원활한 공정 제어를 위해 가상 계측 (VM)의 필요성이 증가

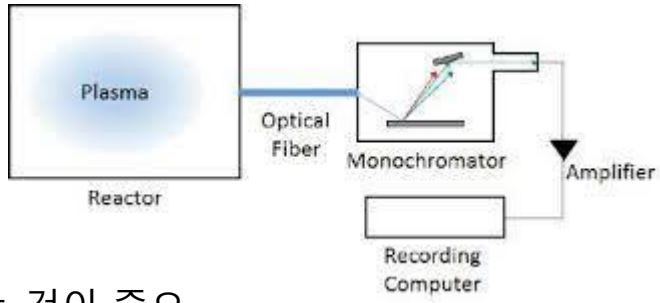
2. 증착 공정 가상 계측 모델링 논문 리뷰

2. Introduction

- 플라즈마 공정 모니터링 센서



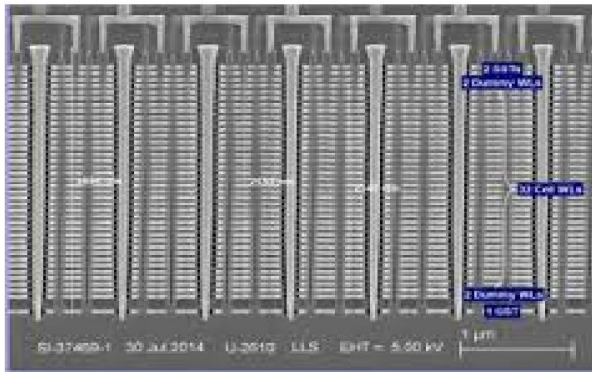
- Optical emission spectroscopy (OES) 센서 : 플라즈마 내의 화학종을 측정하는 센서
- PECVD 증착 공정 가상 계측을 위해 OES 센서를 채택하여 데이터 취득
- OES 데이터의 특징
  - 3차원 데이터 (wavelength, time, intensity)
  - 대부분의 데이터는 dark signa을 가진 noise data
  - Domain knowledge 기반 의미 있는 파장의 데이터를 추출하는 것이 중요



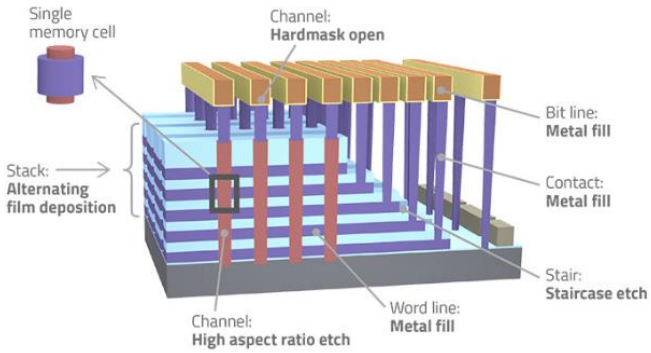
2. 증착 공정 가상 계측 모델링 논문 리뷰

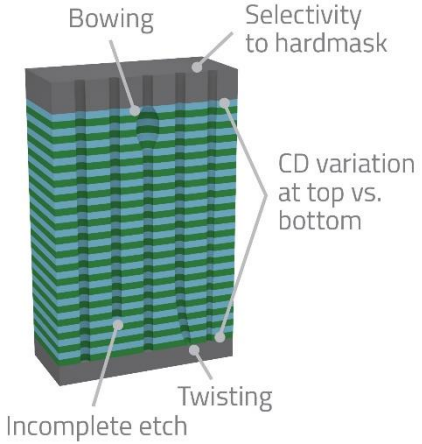
2. Introduction

- Target 공정 : 비정질 탄소막 (Amorphous carbon layer, ACL) 하드 마스크 증착 공정
  - High aspedct ratio contat (HARC) process : 3D NAND Flash memory의 핵심 중 하나
  - HARC 공정을 위해 하드 마스크의 사용이 불가피

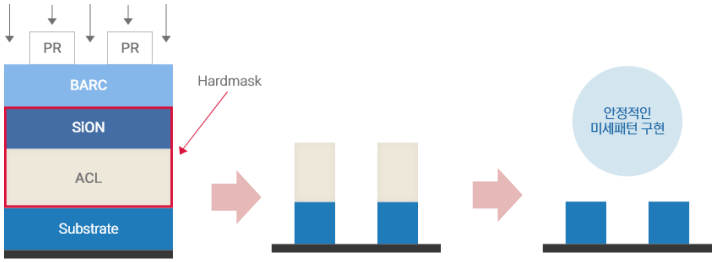


3D V-NAND flash memory HARC process
(Source : Samsung)





Structure of 3-D NANA flash memory cell formation (left) and Various issue of etch process Bowing, Twisting, C variation at op vs bottom



ACL patterning

- 하드마스크 용도로 ACL 박막 활용
- ACL의 두께와 조성에 따라 하드마스크 성능 변화
- ACL이 하드마스크로서 역할을 제대로 수행하지 못할 경우
  => 즉각 공정 이상 초래

→ 원하는 두께와 조성을 얻을 수 있도록 ACL 증착 공정 최적화 필요

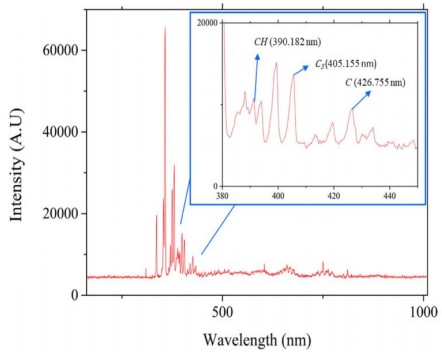
2. 증착 공정 가상 계측 모델링 논문 리뷰

3. Experiment

- 장비 및 실험
  - 13.5 MHz RF powered CCP type PECVD
  - 실험 계획법 (Design of experiments, DOE) : Box-Behnken(4요인 , 3수준)

Input parameters	Ranges	Unit
RF Power	230-270	wait
Pressure	800-1200	mTorr
$C_3 H_6$	60-100	Sccm
N_2	30-50	Sccm

  - 다른 실험 조건은 고정 (공정 시간 5분, Ar flow 4 sccm, chuck 온도 300 ºC)
  - 재현성 테스트를 위한 3번의 동일 공정을 포함한 총 27개의 공정 진행
- OES 측정 데이터



  - 실험 중 취득된 OES full spectrum 확인
  - Domain knowledge 기반 공정 관련 peak 확인
  - CH (390 nm), C_3 $(450\,nm)$, C $(426\,nm)$
  - 넓은 영역의 N_2 band 영역 확인

- 실험 중 취득된 OES full spectrum 확인
- Domain knowledge 기반 공정 관련 peak 확인
- CH (390 nm), C_3 $(450\,nm)$, C $(426\,nm)$
- 넓은 영역의 N_2 band 영역 확인

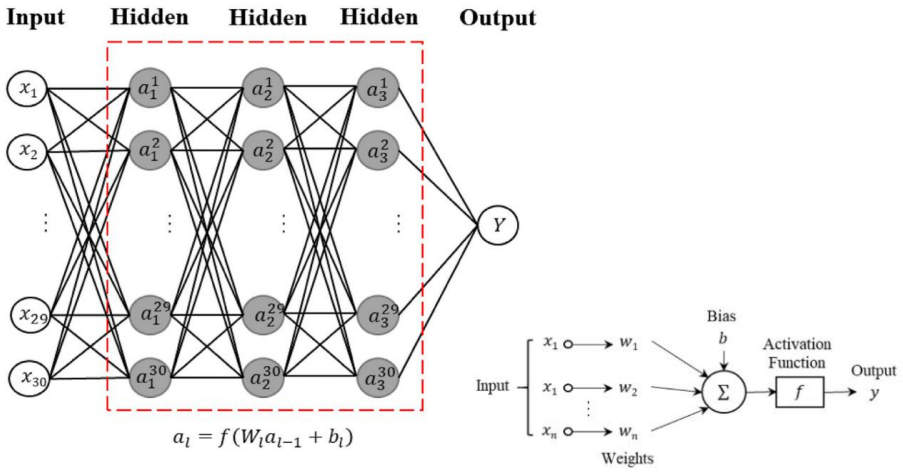
2. 증착 공정 가상 계측 모델링 논문 리뷰

3. Experiment

- Input / Output data

Recipe data	Sensor data	Wafer metrology data
RF Power	OES sensor data	Thickness of the ACL film
Pressure		
$C_3\,H_6$ gas flow		
N_2 gas flow		

- Algorithm : 인공신경망 (Artificial neural network, ANN)



Hyperparameters	Value
Number of hidden layers	3
Number of hidden nodes	30 (Each layer)
Activation function	ELU
Batch size	1 (SGD)
optimizer	Adam

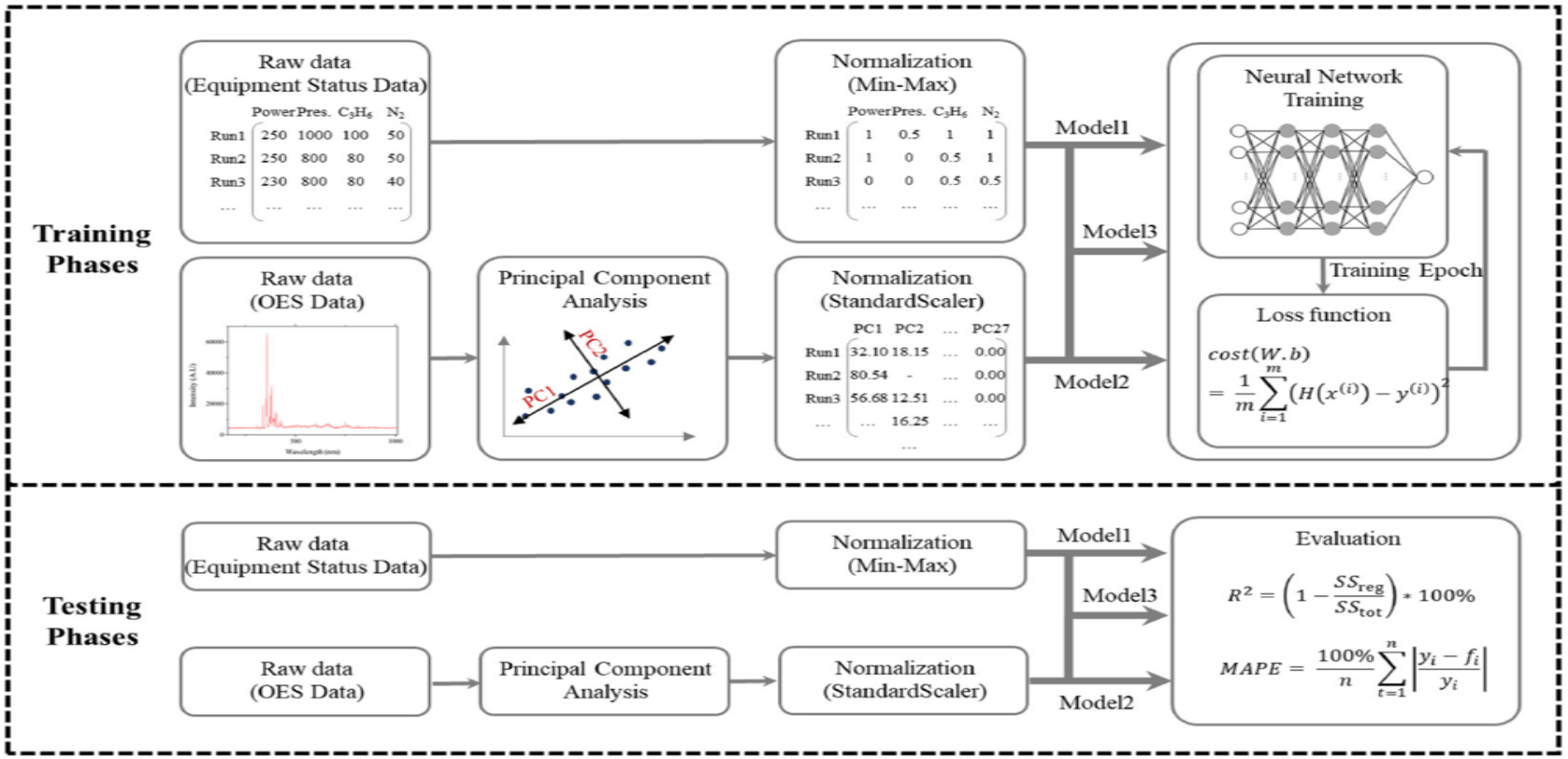
2. 증착 공정 가상 계측 모델링 논문 리뷰

4. Modeling

- Machine learning diagram
  - Recipe data : RF Power, pressure, C3H6 gas flow, N2 gas flow
  - OES data : Full spectrum 3648 wavelengths -> PCA 27

$$\text{Min} - \text{max} \quad \text{normalization:} \quad X = \frac{x - x_{min}}{x_{max} - x_{min}}$$

$$\text{StandardScaler} \quad : \quad X = \frac{x - \mu}{s}$$

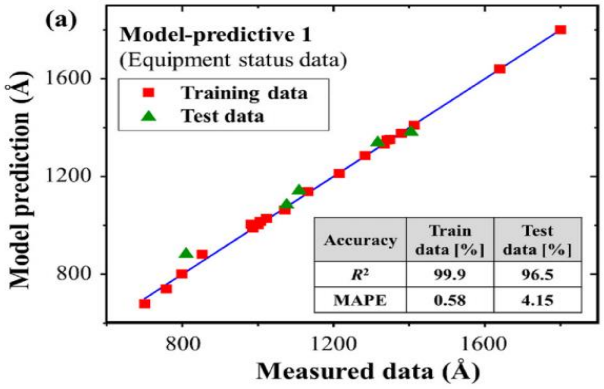


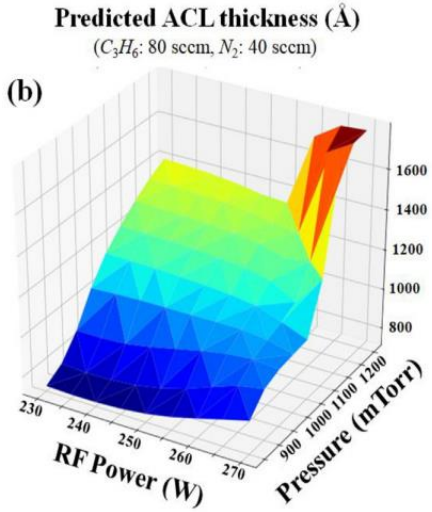
2. 증착 공정 가상 계측 모델링 논문 리뷰

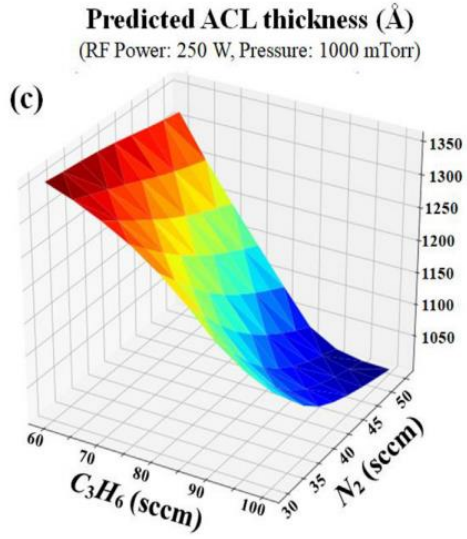
4. Modeling

- Thickness prediction results

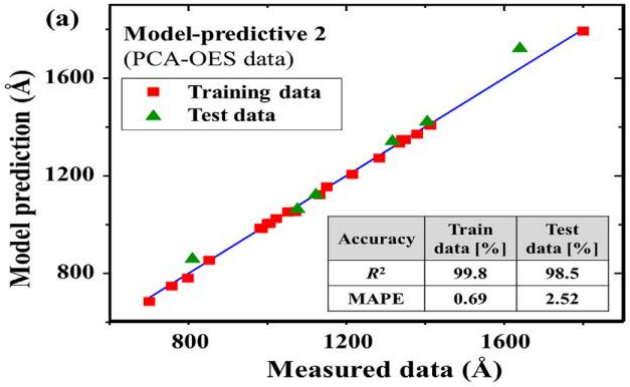
[Model 1] input : Recipe, Output : Thickness



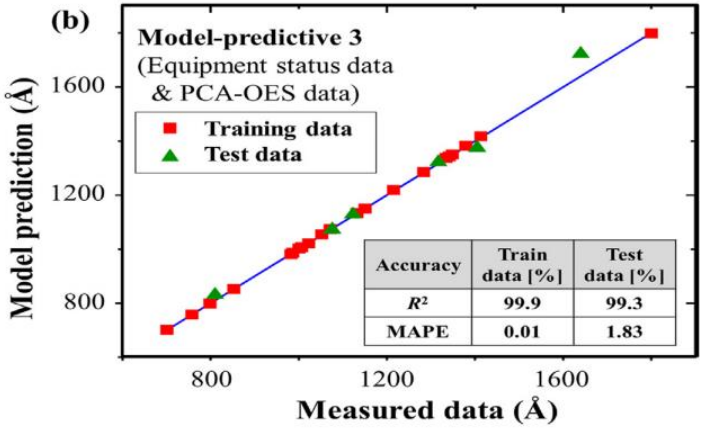




[Model 2] input : OES, Output : Thickness



[Model 3] input : Recipe + OES, Output : Thickness

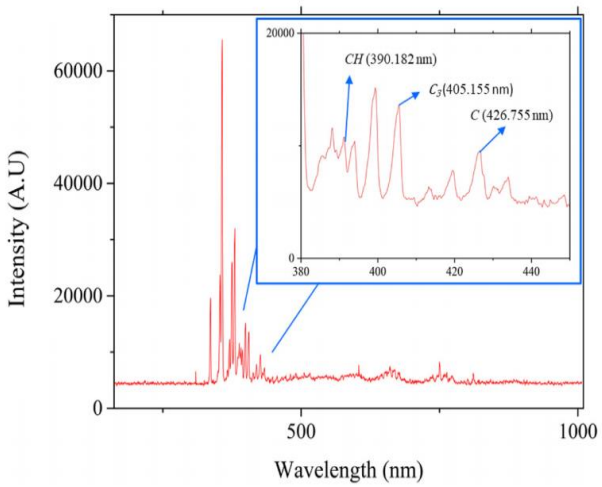


2. 증착 공정 가상 계측 모델링 논문 리뷰

4. Modeling

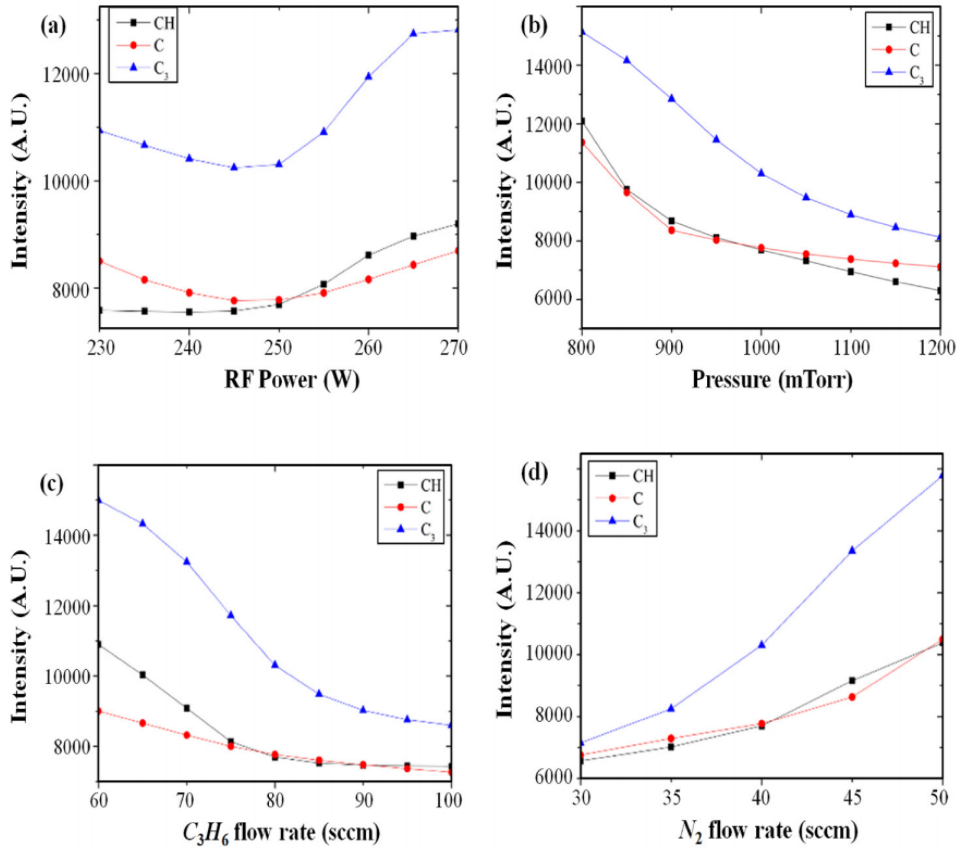
- OES prediction results

Input : Recipe, Output : OES intensity



$$I_x = C_x(E)\eta_x[X]$$

I_x : Emission line intensity
$C_x(E)$: rate coefficient (Constant)
η_x : excitation efficiency = $k_e(E)\ n_e$
$k_e(E)$: excitation rate constant
n_e : electron density



2. 증착 공정 가상 계측 모델링 논문 리뷰

5. References

- Tung-Ho Lin, Ming-Hsiung Hung, Rung-Chuan Lin, and Fan-Tien Cheng, "A Virtual Metrology Scheme for Predicting CVD Thickness in Semiconductor Manufacturing", 2006 IEEE international Conference on Robotics and Automation, p. 1054, May, 2006.
- Pilsung Kang, Hyoung-joo Lee, Sungzoon Cho, Dongil Kim, Jinwoo Park, Chan-Kyoo Park and Seungyong Doh, "A Virtual Metrology System for Semiconductor Manufacturinig", Expert Systems with Applications, Vol. 36, Issue 10, p. 12554, December, 2009.
- Haegyu jang, Kyongbeom koh, Honyoung Lee and Heeyeop Chae, "Plasma Monitoring by Multivariate Analysis Techniques, "Vacuum Magazine, Vol. 2, NO 4, pp. 27-32, 2015.
- Yongsik Yu, "Tech Brief: Memory "Grows Up" with 3D NAND", Lam RESEARCH, http://lamresearch.com, 2016.