Machine Learning Summary

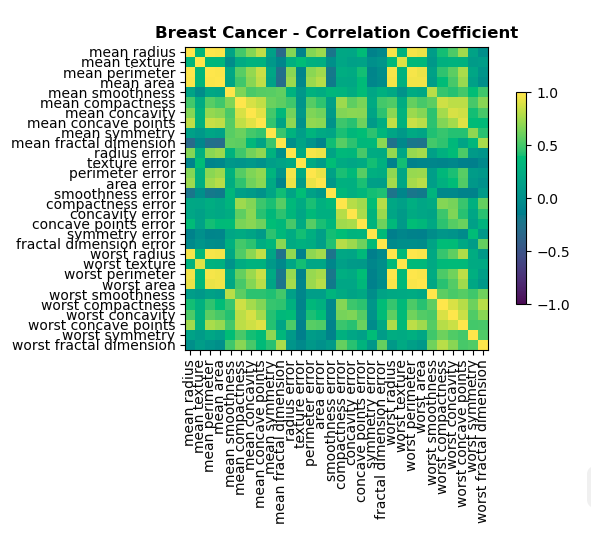
1. **Machine Learning 접근 방법**
   1. **Machine Learning 정의**
      * **Field of study that gives computers the ability to learn without being explicitly programmed (Arthur Samuel, 1959)**
        + 명시적인 프로그래밍 없이 데이터만을 제공하여 경험 누적을 통한 성능 개선
        + Data를 제공하여 문제를 해결하거나 기능을 수행하는 모델을 만들어내는 기법
        + Input : Model Input & Output Dataset
        + Output : Model (Algorithm)
      * **Classical/Conventional Approach (Non-ML / Non-NN)**
        + 목표하는 기능 및 문제에 대해 매번 모델/알고리즘을 재설계해야함
        + 목표하는 기능 및 문제에 대해 모델/알고리즘의 Parameter를 Fine-Tuning 해야함
      * **Machine Learning / Neural Network Approach**
        + 목표 기능 및 문제를 표현하는 Dataset을 이용한 학습을 통해 모델/알고리즘을 도출함
        + Dataset에 추가 데이터를 업데이트하고, 새로운 추가 데이터를 학습하여 모델/알고리즘의 성능을 개선함
        + 모델/알고리즘의 Parameter는 학습을 통해 Tuning 되기에 학습에 대한 Hyper Parameter (반복 학습 횟수 Epoch, 1회 학습 데이터 크기 Batch Size, 학습 속도 Learning Rate 등)을 조정하여 모델/알고리즘의 학습 과정을 최적화함
        + 모델/알고리즘의 성능 개선을 위해서는 좋은 Feature로 구성된 많은 데이터가 필요하며, 목표하는 기능 및 문제에 따라 매번 다른 학습이 요구됨
   2. **Machine Learning 기법 분류**
      * **Supervised Learning**
        + Input Dataset과 Output Label (Input에 대한 정답)을 모두 제공하여 모델을 학습하는 방법
        + Input 데이터에 따라 Error가 최소화될 수 있도록 (Loss Function 값이 최소화도록) 적합한 정답을 도출하도록 모델/알고리즘을 학습시킴
        + 정해진 Label을 추정하도록 학습되기에 Classification 문제를 해결하는 데에 특화됨
        + Linear Regression, Logistic Regression, KNN (K-Nearest Neighbors), SVM (Support Vector Machine), Decision Tree, Random Forest, Neural Network
      * **Unsupervised Learning**
        + Output Label 없이 Input Dataset만 사용하여 모델을 학습하는 방법
        + Clustering, Dimensionality Reduction, Association Rule Learning
      * **Semi-Supervised Learning**
        + Supervised & Unsupervised Learning의 조합으로 이루어짐
      * **Reinforcement Learning**
        + Agent가 상태(State)의 보상 (Reward) 최대화하는 방향으로 Neural Network의 Parameter를 최적화함
   3. **Dataset 구조**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Column 1  (Feature 1) | Column 2  (Feature 2) | · · · | Column M  (Feature M) |
| Data 1 |  |  |  |  |
| Data 2 |  |  |  |  |
| ·  ·  · |  |  |  |  |
| Data N |  |  |  |  |

* Dataset은 NxM 형태의 Matrix로 구성되어있음. 데이터셋은 N의 데이터로 구성되어있으며, 각 데이터는 M개의 Feature로 구성되어 표현됨
* Dataset의 Row Vector는 데이터셋의 데이터 1개를 지칭함 (데이터 N개)
* Dataset의 Column Vector는 각 데이터를 표현하는 Feature를 상징함 (Feature M개)
* Dataset의 데이터는 ML 및 Neural Network 모델에 1xM 형태의 벡터로 입력됨
  1. **Machine Learning Implementation 순서**
     1. **Dataset 준비**
        + 목표하는 기능과 문제와 관련된 데이터셋을 확보해야함
        + 데이터셋은 ML/DL 라이브러리와 연동되기 위해 NxM Matrix 형태로 구성되어야함
     2. **Dataset Preporcessing**
        + 데이터셋의 각 Feature는 다른 단위의 값으로 구성됨
        + 각 Feature별로 다른 단위를 사용하게 되면서 데이터의 분포가 달라지고 학습에 대한 영향력이 달라지는 Scale 문제로 인해 학습 과정 중 Overfitting과 같은 Fitting 문제가 발생함
        + Standardization, Normalization 등 데이터셋의 단위를 통일시키는 Preprocessing을 수행하여 모델/알고리즘의 Fitting을 개선함
        + 만약 현재 데이터셋에서 사용하는 Feature가 너무 많아 Classification을 위한 학습이 복잡할 경우 데이터셋을 더 적은 Feature로 재구성하는 Dimensionality Reduction이 요구될 수 있음. 이 때, PCA (Principle Component Analysis)를 통해 설명력이 높은 소수의 새로운 Feature로 데이터셋을 재구성하여 Classification 학습에 좀 더 쉬운 방향으로 유도할 수 있음.
     3. **Dataset Analysis를 통한 ML 기법 선정**

Machine Learning과 Neural Network는 데이터셋에 매우 의존적이기 때문에 데이터셋의 Feature 성격에 따라 적합한 모델을 선정해야함

* + - * **Multi-Colinearity 존재할 경우**
        + 우선적으로 데이터셋의 Feature간 선형성 (Colinearity)를 파악해야함.
        + 다수의 Feature로 구성된 데이터셋에서 Feature 간에 서로 독립적인 것이 이상적이나 Feature간 서로 영향을 주는 Multi-Colinearity가 존재할 수 있음.
        + 이를 파악하기 위해 Feature M개에 대한 MxM Correlation Matrix를 구성해서 각 Feature가 서로에게 Correlation (선형성 존재여부)을 가지는지 파악함.
        + Multi-Colinearity가 존재하는 Feature 사이에는 Linear Classifier는 적용할 수 없음.

  
Fig 1: Correlation Matrix (Breast Cancer Dataset)

* + - * + Multi-Colinearity가 존재하는 Feature 사이에 Classification을 위해서는 보다 복잡한 구조의 모델/알고리즘이 요구됨.

(ex : Neural Network, SVM, Random Forest, Decision Tree)

* + - * **데이터셋의 Non-Linearity 존재할 경우**
        + 데이터셋 Feature의 물리적 특성이 달라 Non-Linearity가 존재할 수 있음.

(ex : 2D 이미지 vs 3D Translation / 2D 이미지 vs 3D Rotation)

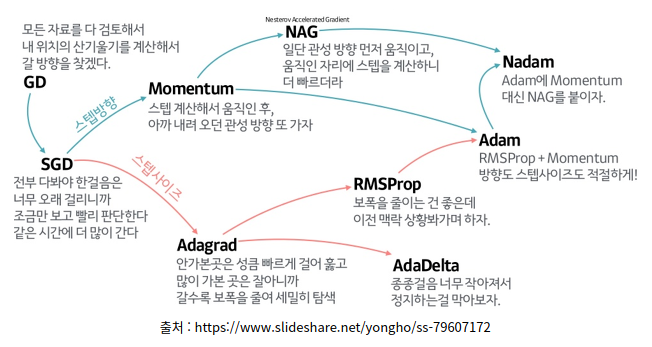
* + - * + Non-Linear 데이터셋에 대해서는 Linear Classifier를 적용할 수 없으며, 더 많은 Parameter를 학습하여 Non-Linear한 추정을 할 수 있는 복잡한 모델/알고리즘이 요구됨. (ex : Neural Network)
    1. **Training/Validation/Test Dataset 분리**

데이터셋은 학습 및 성능 평가를 위해 전체에서 일정 비율로 Training Set, Validation Set, Test Set으로 각각 분리하며, 서로 중첩이 없어야하며 Random하게 선정됨

* + - * **Training Set**
        + ML 모델 학습을 위해 사용하는 데이터셋
      * **Validation Set** 
        + ML 모델 학습에 필요한 Hyper Parameter를 찾기 위해 사용하는 데이터셋
        + 학습이 종료된 후 Test Set이 주어지기 전까지 학습 과정 중 모의 평가를 하기 위해 사용
        + 모의 평가를 통해 Unseen Data에 대한 정확도를 확인하여 Test Set에 대한 정확도를 가늠할 수 있음
        + 모의 평가를 통해 Training Set, Validation Set 2개에 대해 모두 정확도가 높게 나오게 만듬으로서 Training Set에만 최적화되는 Overfitting을 방지함
      * **Test Set**
        + 학습된 모델을 평가하기 위한 데이터셋
    1. **Training 수행 + Optimization**
       - **Loss Function 선정**
         * 목표하는 Output을 최대한 원하는 정답에 가깝게 만들기 위해 Error를 최소화 시키는 방향으로 학습해야함
         * 사용하는 모델의 Output 형태와 Label/Target Output의 형태에 따라 Error의 형태와 종류가 다름. 그러므로 최소화하려는 Error의 형태에 따라 적합한 Loss Function을 선정해서 학습을 수행함.

(ex : MSE (Mean Squared Error), Cross Entropy Error)

* + - * **Optimizer 선정**

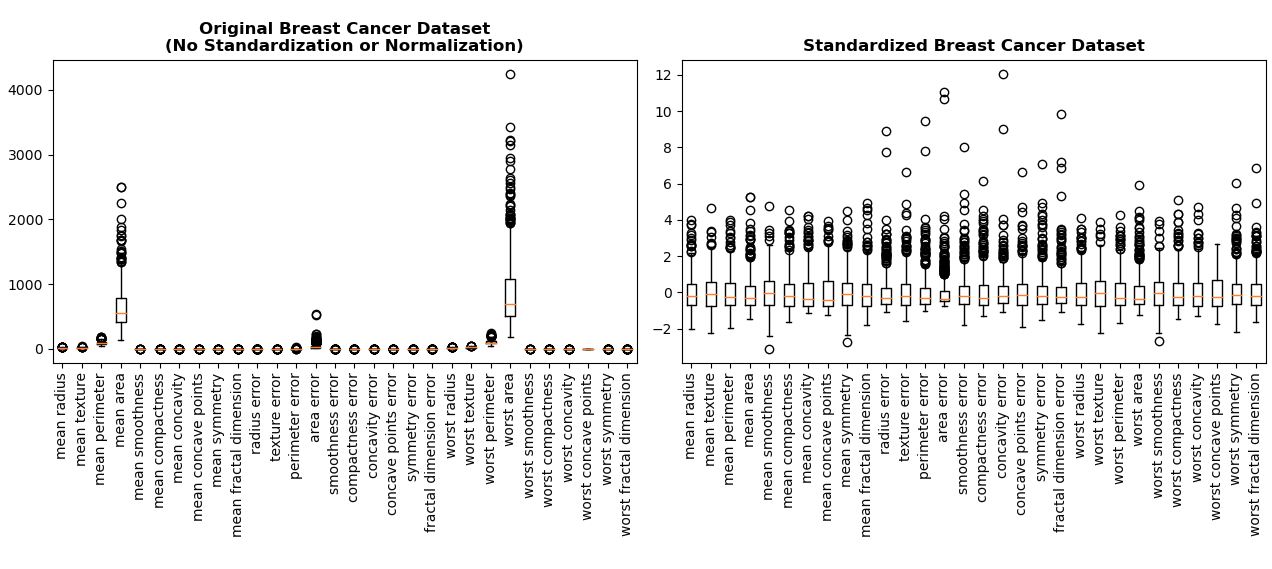
  
Fig 2: Optimizer Types

* + - * + ML/NN 모델 학습 방향과 방식에 따라 Optimizer를 선정하여 사용함
      * **Training 수행**
        + Training 수행 시 Training Set과 Validation Set에 대해 각각 학습을 수행하여 Overfitting을 방지함
        + Training Set에서의 Loss Functin, Validation Set에서의 Loss Function을 각각 확인하여 Overfitting, Underfitting을 판단함
  1. **Machine Learning Dataset Preprocessing 및 Analysis 기법**
     + **Dataset Preprocessing**
       - **Dataset Rescaling**
         1. **Standardization (Z-Score / 표준화)**

**Standardization 공식 및 효과**

|  |  |
| --- | --- |
|  | X' : 표준화된 데이터 (Z-Score)  X : 원본 데이터  µ : 특정 Feature에 대한 X의 평균  σ : 특정 Feature에 대한 X의 표준편차 |

* + M개의 Feature로 구성된 데이터에 대해 각 Feature의 평균과 표준편차를 사용하여 각 Feature에 대해 Z-Score로 데이터를 Rescaling하여 표준화함
  + Standardization을 통해 데이터셋의 각 데이터는 1 x M으로 구성된 Z-Score 데이터로 재구성됨

  
Fig 3: Dataset Standardization (Breast Cancer Dataset)

* + Standardization을 통해 데이터셋의 모든 Feature는 동일한 평균과 표준편차를 가지게 되며, 동일한 범위의 분포 내에 재구성될 수 있음
  + Standardization을 통해 각 Feature별로 다른 Scale을 가지는 것을 동일한 분포를 가지게 하여 Scale을 통일시킬 수 있음

**사용 API**

**Scikit Learn – StandardScaler의 fit 함수, transform 함수**

(sklearn.preprocessing.StandardScaler)

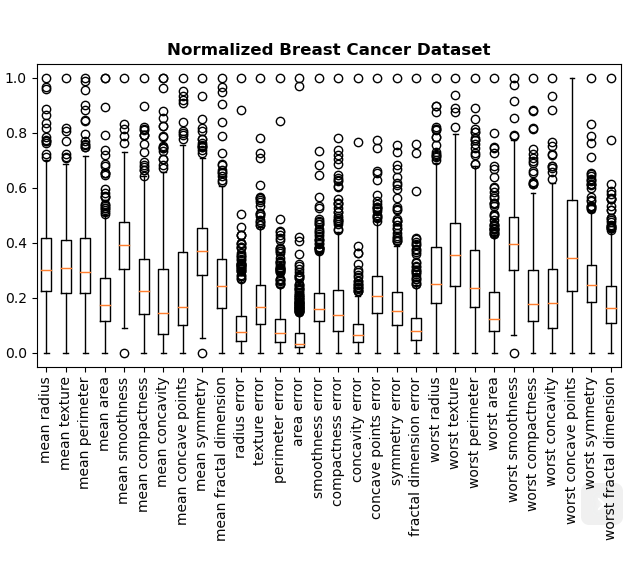
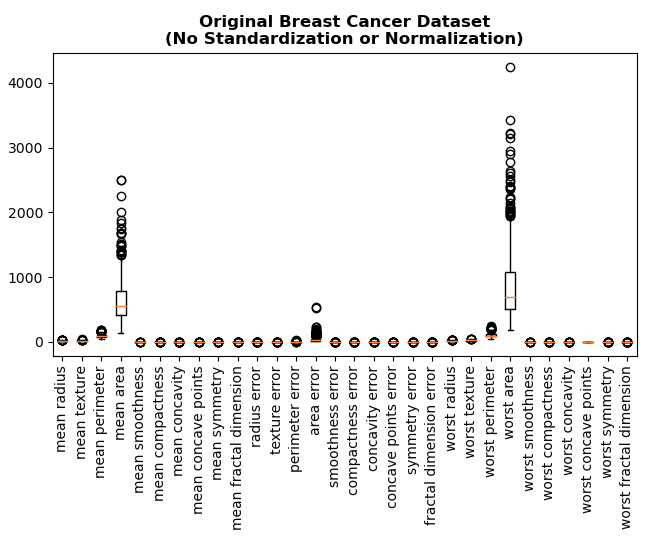
fit 함수를 통해 평균과 표준편차를 구하고, transform 함수를 통해 Feature 데이터를 Z-Score로 표준화함

* + - * 1. **Normalization**

**Min-Max Normalization 공식 및 효과**

|  |  |
| --- | --- |
|  | X' : 정규화된 데이터  X : 원본 데이터  max(X) : 특정 Feature에 대한 X의 최대값  min(X) : 특정 Feature에 대한 X의 최소값 |

M개의 Feature로 구성된 데이터에 대해 각 Feature의 최대값과 최소값을 사용하여 각 Feature에 대해 0 ~ 1 사이의 값으로 Rescaling하여 정규화함



Min-Max Normalization은 데이터셋의 각 데이터를 0 ~ 1사이 값으로 구성된 1 x M 데이터로 재구성함

Min-Max Normalization을 통해 각 Feature별로 다른 Scale을 가지는 것을 0 ~ 1 사이 값으로 통일시킴으로서 Scale을 통일시킬 수 있음

**사용 API**

Scikit Learn – MinMaxScaler의 fit 함수, transform 함수

(sklearn.preprocessing.MinMaxScaler)

fit 함수를 통해 최소값과 최대값을 구하고, transform 함수를 통해 Feature 데이터를 0 ~ 1 사이로 정규화함

* + - **Dimensionality Reduction (차원 축소)**
      1. **Simple Projection**
      2. **PCA (Principle Component Analysis)**
    - **Dataset Analysis**
      * **Linearity 판단 : Correlation Matrix**
  1. **Machine Learning 설계 주요 중점**
     + **Overfitting**
       - **발생하는 이유**
       - **해결방법**
     + **Underfitting**
       - **발생하는 이유**
       - **해결방법**

1. **다양한 기초 Classification 기법**
   1. **Linear Regression**
      * **주요 특성 및 원리**
      * **한계점**
   2. **Ridge / Lasso / ElasticNet**
      * **주요 특성 및 원리**
      * **한계점**
   3. **KNN (K-Nearest Neighbors)**
      * **주요 특성 및 원리**
      * **한계점**
   4. **Logistic Regression**
      * **주요 특성 및 원리**
      * **한계점**
   5. **Naive-Bayes Classifier**
      * **주요 특성 및 원리**
      * **한계점**
2. **SVM (Support Vector Machine)**
   1. **주요 특성 및 원리**
   2. **한계점**
3. **Decision Tree**
   1. **주요 특성 및 원리**
   2. **한계점**
4. **Random Forest**
   1. **주요 특성 및 원리**
   2. **한계점**
5. **Optimization : Gradient Descent**
   1. **최적화 원리 및 방식**
6. **ANN (Artificial Neural Network)**
   1. **ANN 주요 특성 및 원리**
      * **Backpropagation**
   2. **Perceptron**
      * **주요 특성 및 원리**
      * **한계점**
   3. **MLP : Multi-Layer Perceptron**
      * **주요 특성 및 원리**
      * **한계점**