인공지능 기초/응용 sw강좌 머신러닝 프로그래밍(1일차 -1교시)

Python, a tool for AI
- 다른 사람들이 많이 파이썬을 인공지능에서 사용하기에 사용. 이번 강의는 파이썬 자체에 대한 강의는 아님.
- 파이썬은 쉽고, 강력하다.
- 교육용 : 환경이 동일한 것이 중요하다.(구글 - colab) 사용

-multi-purpose(general prupose : GUI, Web, Scripting, etc...) : matlab은 공학 계산용

Cyber Physical System(CPS)
- A tight conjoining of computational and physical resources.
- many applications are implemented in this frame by combining many different technologies

AI(Artificial intelligence) > Machine Learning > Deep Learning
- data-driven deep learning : Machine Learning
- 인공지능, 빅데이터를 한다는 것은 deep learning에 대한 내용

data란 무엇인가?(observed random data)
- learning is based on data.(관측되는 데이터)
- 의도적으로 생성한 것이 아닌 데이터를 의미

예) 대한민국의 평균 키(남녀노소 상관없이 키 측정) :
데이터 -> 데이터는 모집단이라는 가상의 집단이 있다.
(무한히 많은 데이터 존재 : The population includes infinite numbers of data samples that have occurred in the past, are occurring now, and will occur in the future)

모집단에는 대한민국의 성인 몇 천 만 명 중, 일정 분포(확률 분포)를 갖고 있다. -> 랜덤하게 데이터 뽑을 경우 (All observed data are random values extracted from the population), 평균, 분산이 특정 값으로 존재.
- 데이터는 매번 변한다.(If data are collected once again, samples with similar properties but different values are collected)

EDA, 학습론, model selection : 수업 진행

model selection
- validation approach, cross-validation

- 파이썬 라이브러리 : TF, Scikit-Learn, SciPy, Stats Model >Numpy, Pandas> Core(파이썬 기본 문법)
matrix(행렬, 벡터, tensor 등) 같은 것은 core인 파이썬 문법만으로는 부족. 따라서 숫자를 다루기 위해 사용하는 라이브러리 : Numpy

SciPy : linear algebra, 푸리에 변환 등 공학계산용(Sci : science) -> 기계 학습의 모델을 사용하기 쉽게 만든 것이 Scikit-Learn(기계학습에서 사용하는 tree 모델 사용)

딥러닝 : 계산양이 많고, 기존의 표현 방법으로는 한계 도달. 따라서 새로운 플랫폼이 필요하여, TF(텐서플로우) 개발(딥러닝용 플랫폼) - Numpy 기반
: TF(tensorflow) > Numpy > Core

data 기반 분석 : 통계
- 통계에서도 컴퓨터 개발과 함께 언어, 통계방법이 새롭게 발전. -> R 언어 : 통계용 프로그래밍 언어

Numpy 문제점, 한계점 도달 : 다 숫자로만 구성.
- 통계적 R언어의 특징을 파이썬에서 다루기 위해서, Pandas 사용. -> 통계적 분석 방법 사용한 것이 : Stats Model (R의 기능이 옮겨오고 있다. 점점 R을 대체하고 있다.)
- 통계를 위해서, 굳이 R을 사용할 필요는 없다.

이희성) 통계에서는 딥러닝을 시작하기 전에 R이라는 언어를 만들어서 사용하고 있었다. 우리가 딥러닝에서 하려는 분석들을 R을 통해서 진행할 수 있었다. 초기에 python에서 numpy에 문제가 많았다. numpy는 숫자를 기반으로 만들어진 알고리즘이라 남자와 여자를 1과 2로 따로 mapping하고 있던 와중에 데이터가 많아지고 복잡해지는 과정을 거치다 보니 바로바로 매핑하는 데이터시스템이 필요하였고 통계학분야에서 주로 사용한 R에서는 이미 이 문제를 해결하여 사용하고 있는 것을 Pandas라는 형태로 python으로 가져오게 되었다.

2교시

Exploratory Data Analysis : EDA

- see how data looks like
- 데이터가 어떻게 생겼는지를 보는 단계
- 분석 시 처음보거나, 익숙하지 않은 데이터를 주로 다룬다. => 처음 보는 데이터에 대해서 익숙해져야 한다.
- 연구 시 EDA의 중요성이 낮을 수 있다.

전통적인 방식 : 데이터에 대한 이해도는 높다.(전반적인 구조, 변수들의 의미 등) - 그러나 요즘은 data-driven deep learning의 경우, 관계식에 대한 이해가 부족한 편이다. -> sol) EDA

EDA : play with the data before the actual analysis

- data driven analysis : analysis with no prior assumption

EDA often includes 1) checking data distributions = univariate & joint.(데이터 분포 파악) : 히스토그램
2) visualization : scatter plot, heat map with /without dimension reduction
3) Data transformation : raw or logged? -> [1] feature engineering
4) Outlier detection
5) Missing imputation : missing(결측치)

A : 100, B : 120, C : 3, D : 15
C+D => other : 18, 기존의 class 4개가 3개로 바뀌는 것이 feature engineering.

p.16) Data Distribution Check

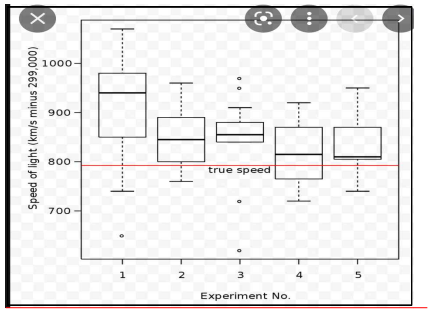
categorical variable : frequency
iris data set sepal length/width, petal length/width, species로 구분

분포값 요약 : ex) Sepal.Length
Min, 1st Qu. Median, Mean, 3rd Qu, Max 등

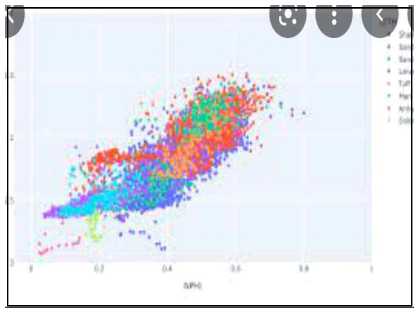
Visualization

- Univariatve visualization(변수 1개를 시각화 : 히스토그램 = one-by-one)

예) histogram of sepal.Length, boxplot



box plot의 예

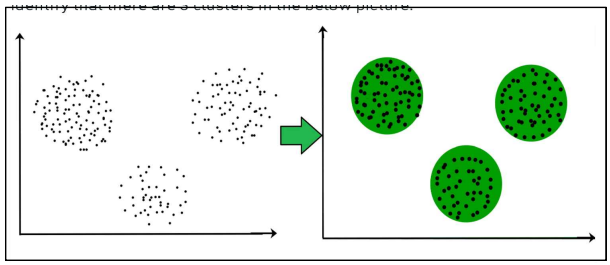


변수 여러 개일 때는 scatter plot 사용

outlier detection : outlier = strange or extreme values in the data(예상치 못한 이벤트에 의해 초래 된 데이터) : can be caused by errors, mistakes, or unusual events, May distort analysis, or can be a new discovery(outlier는 결과 (data)값을 왜곡한다.), 무엇이 outlier라고 명확히 정의내리기 어렵다.(no concrete definition)

outlier are usually detected by

1) physically non-sense data
2) Far away from the center point
(data가 나오지 않을 것 같은 곳에서 생성된 데이터로, 결과 값에 악영향을 준다.) => box plot 범위 밖에 데이터 존재 시 outlier가 된다.



Outlier Detection : multivariate outlier detection

- Clustering-based methods

Missing imputation : 빠지는 값.

- 제거하는 방법도 있지만, 적당한 값으로 채워주는 방법도 있다. -> 만일 너무 많이 제거하면, power가 낮아지기 때문에, 제거하는 대신 채워주는 방법 사용.
-

예) 부모님 학력조사 : 쪽팔려서, 고학력층의 데이터만

---

1) Feature Engineering은 모델 정확도를 높이기 위해서 주어진 데이터를 예측 모델의 문제를 잘 표현할 수 있는 features로 변형시키는 과정이다.
'머신러닝 알고리즘을 작동하기 위해 데이터의 도메인 지식을 활용해 feature를 만드는 과정'이라고 할 수도 있다.
Feature Engineering is a Representation Problem.
Feature: An attribute useful for your modeling task

있을 경우, 모델 만들 경우 왜곡 발생, 모델의 편향성.

PCA(principal component analysis) : 주성분 분석

- 시각화할 때는 차원을 축소할 필요가 있다.
- explaining data : explaining how data vary or data variance, which means information is in variance.

간단하게 차원을 낮추는 방법 2차원에서 x 혹은 y축 중 하나를 버린다.(we want to use only a few variables to explain the data variance as much as possible)

-> x축 혹은 y축 중 하나를 drop할 경우, 50%의 variance 혹은 information 손실 발생.(If we drop X2 and select X1, we lose 50% of variance or information.)

- If we use transformed coordinate of Z1 and Z2, dropping Z2 loses only 20% of variance.(즉, 회전 변환 시 데이터의 손실이 drop할 때 보다 덜 lose.)

=> 최대한 데이터를 유지하면서, 분석하는 것이 PCA

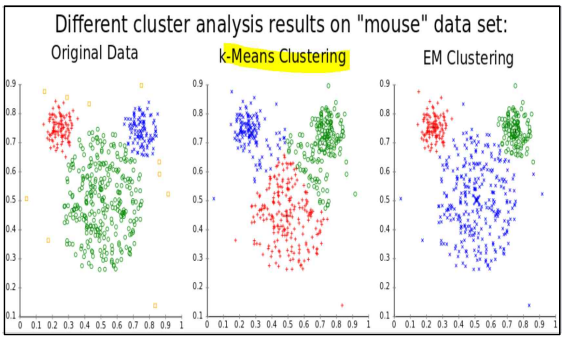
100차원 데이터(변수) - EDA를 위해 모든 데이터를 분석하기 어렵다. => 분석하기 쉽게 3차원으로 변경. (정보 손실 발생), PCA를 사용하여, 데이터의 손실을 최소화하면서, 변경한다.

principal components : orthogonal variables corresponding to the maximum variance, or explaining largest variance.

고차원의 데이터를 저차원의 데이터로 환원시키는 기법을 말한다. (https://ko.wikipedia.org)

clustering : a very broad set of techniques for finding **subgroups** or **clusters** in a data set.

- Grouping samples similar together
- We need to define what is similar or different.
- Market segment : identifying target customers of sale.
- PCA uses a low-dimensional presentation : p -> 2~3.
- **clustering uses a few groups to present samples : n -> 2~3.**



클러스터 분석이란 주어진 데이터들의 특성을 고려해 데이터 집단을 정의하고 **데이터 집단의 대표**할 수 있는 대**표점**을 찾는 것으로 데이터 마이닝의 한 방법이다. **클러스터**란 **비슷한 특성을 가진 데이터들의 집단**이다. 반대로 데이터의 특성이 다르면 다른 클러스터에 속해야 한다.

K-means Clustering : A heuristic algorithm to find a local optimum clustering.
(K=3이면, k개의 cluster 생성.)

한 점(sample)이 k개의 cluster 중 가장 가까운 cluster에 속하게 된다.

4교시

linear model을 주로 설명하는 '수업'.
linear model : f(x) is a kind of linear functions. ex) f(X)=wx+b, A typical parametric model(**선형적 관계를 갖는다.**)

- Linear regression : a linear model for a regression problem.
- Logistic regression : a linear model for a classification problem. => class를 분류할 때 사용.

linear regression : $Y \approx \beta_0 + \beta_1 X$
(one output variable and one input variable)

intercept : 절편, coefficient : 기울기, 예측 값과 실제 결과 값의 차이가 오차

Least square : estimating β_0 and β_1 by minimizing error squares $L(\hat{\beta_0}, \hat{\beta_1}) = \sum_{i=1}^{n} (e_i)^2$: Loss function

easily solve it by differentiating the loss function.

Solution : $\hat{\beta_1} = \dfrac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$

Note that it is minimizing the training error, not necessarily minimizing the test error.

Multiple Linear Regression : $Y \approx \beta_0 + \beta_1 X_1 + \ldots \beta_p X_p$
$\hat{\beta_0}, \hat{\beta_1}$: 예측값

matrix form

$y = \begin{bmatrix} y_1 \\ . \\ . \\ . \\ y_n \end{bmatrix}$, $y = x\beta + e$, $x = \begin{bmatrix} 1\ x_{11} \cdots\cdots x_{1p} \\ . \\ . \\ 1\ x_{n1} \cdots\cdots x_{np} \end{bmatrix}$

Regression Model Evaluation

- MSE(Mean Square Error) : $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$,
- RMSE = $\sqrt{MSE}$
- R^2=r-squared or coefficient of determination. $R^2 = 1$-(unexplained variance) / (total variance of Y), It is usually between 0 and 1, but can be negative if the model is very bad. => 0에 가까울수록 나쁘다. 값은 확률, 비율이기에 0~1의 사이의 값이다.

Logistic Regression : consider a simple binary output Y(0 or 1) and one input X.
Let p(x)=Pr[Y=1|X]. -> class 분류하기.

people like a linear model : $p(X) \approx \beta_0 + \beta_1 X$

The probability should be 0 ~ 1.

$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$, $\log(\frac{p(X)}{1-p(X)}) = \beta_0 + \beta_1 X$

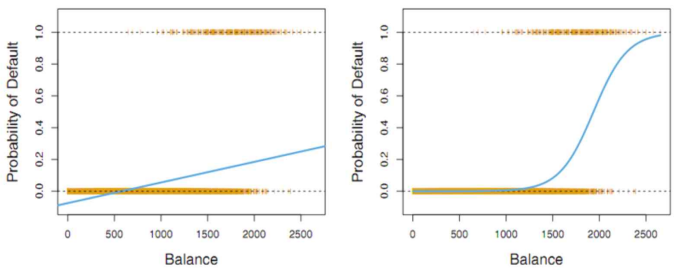


FIGURE 4.2.
왼쪽: 선형 회귀분석을 사용한 확률추정 몇몇의 추정된 확률이 음(-)의 값이다!
오렌지색 막대는 0/1 값을 나타낸다(No 또는 Yes).
오른쪽: Predicted probabilities of default using logistic regression.
모든 확률값이 0과 1 사이에 있다.

오른쪽 : curve 형태. sigmoid function. 0과 1사이의 결과 값을 왔다 갔다 한다.

Likelihood : 지금 내가 보고 있는 데이터를 볼 확률을 최대화 하는 것이 Maximum Likelihood.
-> cross entropy loss.

$\log(\frac{p(X)}{1-p(X)}) = \beta_0 + \beta_1 X$
오른쪽 : linear regression model.

성능평가 : 얼마나 좋은 모델인지에 대한 평가.

-> score 측정 : ACC = MSE = $\frac{1}{n}\sum_{i=1}^{n}(y_i \neq \hat{f}(x_i))$:

The performance of a classification model can be evaluated simply with accuracy or mean square error.

- confusion matrix[2] : a table of the number of samples of true and predicted classes.
(어느 부분이 잘 맞는지, 아닌지 파악가능)

Binary classification Model Evaluation

	Truly positive	Truly Negative
predicted positive	TP	FP

Predicted Negative	FN (False Negative)	TN

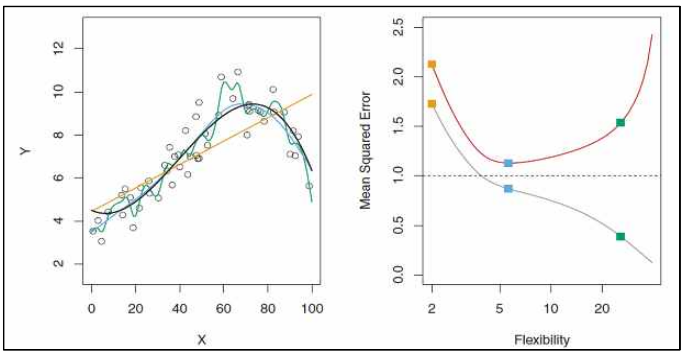
얼마나 정밀하게 예측 했는가? : **precision**, positive predictive value(PPV) = TP/(TP+FP)

False positive Rate(FPR)=FP/(FP+TN)

Recall, True Positive Rate

F1 Score = 2*(Precision * Recall)/(Precision + Recall) : A kind of mean(harmonic mean) of precision and recall
- The ratio of TP to the average of FP and FN.

5교시 : The goal of statistical learning
we know nothing about the test set when we estimate f(X).
How to estimate the test errors without looking at the test set?



test와 train의 Mean Squared Error 값 차이 발생.
-> under fitting, over fitting 발생.

모델이 너무 복잡하면, 새로운 데이터에서 예측할 수 있는 성능이 떨어져 over fitting이 발생한다.

Validation Set : **Pseudo training**을 **training set**으로,

2) **confusion matrix** : 기계 학습 분야의 통계적 분류 같은 문제에서 컨퓨전 행렬, 이란 지도 학습으로 훈련된 분류 알고리즘의 성능을 시각화 할 수 있는 표이다.
행렬의 각 **행**은 **예측 된 클래스**의 인스턴스를 나타내며 각 **열**은 **실제 클래스**의 인스턴스를 나타낸다. 위키백과

pseudo test를 validation set으로 설정.

Final Assess= |Y- $\hat{Y}$|

Preventing Over fitting : A general technique to control the model flexibility, which can be applied to most learning methods $Y \approx f(X|\theta,\lambda)$
- Model parameter θ : parameter obtained from data(ex : coefficient of linear regression models)
- Tuning parameter λ : design choices(K of KNN)
-> 적절한 값을 고를 수 있는 데이터(choices)

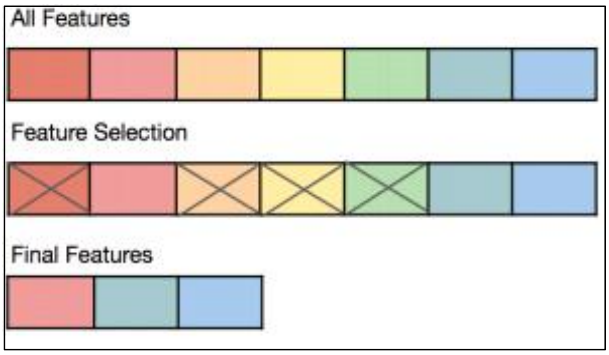
under fitting보다 over fitting이 심각한 문제. -> under fitting 발생 시 simulation이 동작하지 않지만, over fitting의 경우는 동작하는데, 결과 값에서 오류가 생겨, 원인 모를 결과 값 오류로 고생한다.

대부분의 모델에 적용가능한 방법
1) Feature Selection : Among ρ input variables X's, selecting the best K variables.
(가장 성능이 좋은 K개만 선택해서 사용한다.)

2) dimension Reduction
- Reducing the p-dimensional input data space into an k-dimensional space.
- similar to the feature selection, but usually the feature information is not preserved.
(핵심 : 몇 차원으로 줄일 것인지)

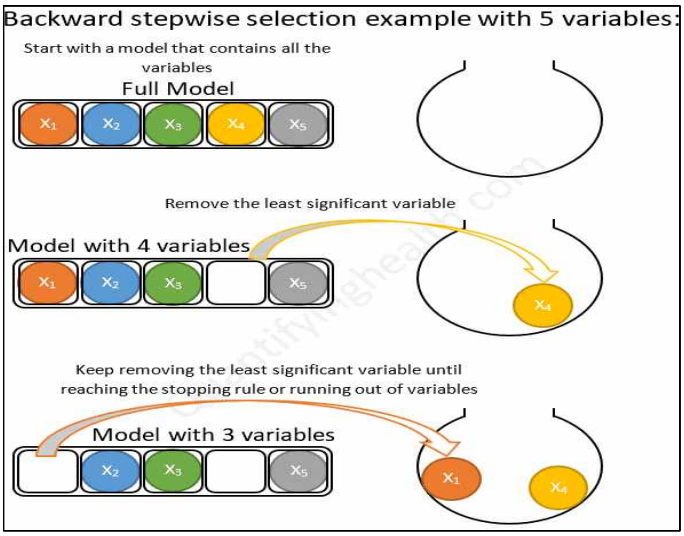
3) Penalization : limit the range of model parameters => 모델의 simple화가 가능하다.

Feature Selection : selecting k best predictors among p predictors. (k개의 가장 성능이 좋은 것을 선택한다.)
- best : the lowest test MSE.



Feature Selection을 보면, All Features와 달리 일부 값을 "x"로 제외한다는 것을 알 수 있다.

Stepwise Selection



backward selection은 처음에 모든 값을 입력하고, 성능이 안 좋은 것들을 반복하면서 뺀다.

dimension reduction

$$Z_m = \sum_{j=1}^{p} \Phi_{jm} X_j$$

penalization : 모델 파라미터의 범위를 제한함으로써 모델의 복잡성을 조절하는 방식 -> 현재 가장 많이 사용되는 방법.
β^*=argmin(Error + λPenalty)
Ridge : β^*=argmin(RSS + $\lambda\sum\beta_i^2$),
$\lambda\sum\beta_i^2$: penalty term

Lasso : β^*=argmin(RSS + $\lambda\sum|\beta_i|$)
Elastic Net : β^*=argmin(RSS + $\lambda_1\sum|\beta_i| + \lambda_2\sum\beta_i^2$)

범위를 제한하고, argmin으로 최소화를 함으로써 결과적으로 모델의 복잡성을 조절한다.

교차검증, over fitting 방지를 위한 tech을 학습함.