**고려대학교 전기전자공학부 석준희 교수님**

데이터분석(빅데이터) ==> 인공지능(기계학습) / 머신러닝은 8월 과정

인공지능 == 빅데이터, 하나의 연장선이라고 볼 수 있음. 결국엔 비슷한 내용을 다룸

인공지능, 1960년대에 나온 오래된 개념

빅데이터, 2000년대 이후에 나온 개념

* [데이터 - 수집, 처리, 저장 기술]; 현재 공공 빅데이터 등은 저장, 수집에 초점이 맞춰있음.(데이터 엔지니어링)

빅데이터 플랫폼을 구축하지 않고 기존의 데이터를 가지고 빅데이터로 활용하는 것은 어려움

=> 기존의 데이터는 축적시에 재활용을 목적으로 축적하지 않고, 업무 활용성을 위해 그냥 저장하니깐

=> 빅데이터를 축적하기 위해서는 저장 방식, 플랫폼, 시스템 등을 정교하게 구축해야 가능함

* [데이터 - 분석, 지식 추출 기술](데이터 분석, analytics)

인공지능에서의 확률론적인 결정

X(environment) > Machine > Y(action)

규칙 기반의 알고리즘

* 지식 경험 기반의 규칙
* 데이터 기반의 규칙(우리가 해야할 것)

통계분석 => 기계학습 => 딥러닝

IBM 왓슨 => 규칙기반의 인공지능

하지만, 룰이 애매한 문제의 경우에는 컴퓨터로 구현하기가 어려움

데이터 기반의 인공지능(기계학습)

**딥러닝, 기계학습, 인공지능 vs 빅데이터**

* **빅데이터에는 데이터 수집, 저장, 공유라는 고유의 영역이 있음(빅데이터 초기)**
* **데이터를 활용한 분석(빅데이터 최근)**

우리가 접하는 대부분의 문제는 스몰데이터만으로도 해결이 가능함.(통계분석, 기계학습)

빅데이터만으로 해결할 수 있는 문제와 가치를 찾아내는 것이 중요함.(최근엔 딥러닝)

**Python**

스택구조

Core(파이썬 기본 구조) > Library(Numpy, Pandas, Scipy, tensorflow 등) >

파이썬의 기본 구조 중에 리스트가 있지만, 리스트의 경우에는 데이터가 많아질 경우 메모리를 많이 사용하게 됨.

* Numpy : 숫자만을 효율적으로 사용하기 위한 데이터 타입 라이브러리
* Pandas : 다양한 데이터 타입, 데이터 확장을 용이하게 하기 위한 라이브러리
  + 문제는 Pandas는 나중에 개발된 것이기 때문에 Scipy와 Scikit-learn과 호환이 안될 경우가 있음
* Scipy : 통계기법을 효율적으로 사용하기 위한 라이브러리
* Scikit-learn : 기계학습(선형회귀, 로지스틱 회귀, 랜덤포레스트) 등을 포함
* Stats Models : 통계분석을 전문적으로 하기 위함(R의 기능을 가져오기 위해 만든 라이브러리)

메모리를 효율적으로 사용하기 위해서는 숫자로 변환하는 과정이 필요한데, 문제는 숫자로 변환할 경우 데이터 의미 손실 혹은 의미 유추 등이 어려워짐

* R의 데이터 구조를 참고삼아 Pandas 사용

⇒ 딥러닝은 위 라이브러리 만으로는 계산할 수 없음 … tensorflow!

**확률**

Sample space : 실험에서 가능한 모든 결과의 집합 ex) 주사위 {1,2,3,4,5,6}

Event : 표본공간의 부분집합( 일부 결과물) {1}, {1,2}, {2,4,6} ...

확률 : a mapping function of event to a real number(0~1) ; 확률은 하나의 함수임

ex) 동전 앞면이 나올 확률(함수) 0.7 이라고 맵핑할 수 있지만, 잘못된 함수인 것임

해당 함수는 확률공리라는 것을 만족해야 함

확률변수(random variable) : X ( continuous , discrete); 어떠한 값이든 가질 수 있는 추상화 값

* continuous : 연속 => pdf
* discrete : 이산 => pmf
* numeric : 숫자
* categorical : 범주

discrete 중에서 numeric이 아닌 것도 있음 (0 , 1 -> 남자, 여자일 경우)

연속확률변수의 특정값일 확률은 0

Pr[X=47] = 0 ; Pr[45 < X < 47] = 0.9 ;적분

확률밀도 함수(f(x)) : Pr[x<X<x+dx]/dx; f(x)는 사실 확률이 아니라 확률을 확률의 공간으로 나눈값

[자연 현상의 분포들]

현실에서 일어나는 사건들 중에서 어떠한 조건들을 만족하냐에 따라서 분포가 나뉨

대표적인 것이 정규분포(자연 현상에서 가장 많이 볼 수 있는 분포)

* 실제로 정규분포를 따를 경우 통계적으로 적용할 수 있는 방법들이 많음.
* 정규분포를 따를 때 적용할 수 있는 통계적인 기법과, 정규분포를 따르지 않을때 적용할 수 있는 방법론적인 것을 이해하고 있어야 함.

최근에는 모수적 기법을 가정하지 않고서 비모수 기법으로 문제를 해결할 수 있는 방법들이 많으 나옴 (데이터 관찰 개수가 많아졌기 때문)

**우리는 표본을 가지고 어떤 조건을 만족하면 어떤 분포를 따를 것이다라는 가정으로 분석을 시작함.**

그렇다면, 어떤 분포를 가정하느냐에 따라 적용할 수 있는 방법들이 달라질텐데, 그런 방법론적인 문제는????

신용카드 사용액은 정규분포가 아닌 꼬리가 긴 로그노말 분포다. 어떠어떠한 이유 때문에

**중심극한정리 => 모집단의 평균과 분산이 궁금할 때 유용**

우리가 표본을 뽑아서 그것을 모집단의 평균으로 가늠해볼 수 있는 이유는, 중심극한 정리 덕분

지금까지는 확률변수 X 한 개에 대해서만 다룸

그렇다면 확률변수 Y와 확률변수 X를 다루기 위해서는 결합확률 분포를 써야함

하지만, 공분산을 통해 결합확률 분포를 대체할 수 있음.

공분산 => 확률변수 2개의 관계를 나타내는 값

* 확률변수 X와 Y가 함께 변화하는 정도를 나타냄
* Y: 몸무게 / X : 키, IQ ; 키와 IQ중에 몸무게와 더 관련 깊은 것은 어느 것인가?
* 관련 높은 것끼리 변화량(+양의관계, -음의관계 ; 절대값이 중요) 이 더 비슷함
* 단점 : 데이터의 스케일에 따라 값이 달라질 수 있음(실질적인 의미는 변하지 않아도)

상관관계 계수(-1 to 1) : 공분산을 각각 확률변수의 표준편차로 나눈 값

* 공분산에서 있는 단위 스케일의 취약점을 보완함
* 확률변수 X와 Y가 같이 얼마나 변하는지 나타내는 정도
* 단점 : 선형관계만 표현할 수 있음

조건부확률

* 우리가 보는 변수의 값 중에서 특정 값으로 범위를 좁힐 때 사용

독립

* 확률 변수 X1, X2과 관련이 있는가 없는가?

독립이면 COV=0 (같이 변화하지 않으니깐), COV=0 이라고해서 독립을 의미하는건 아님

Y와 변수들의 관계가 독립이면 해당 변수는 Y을 예측하는데 없애도 됨

(Corr이 0이면 독립을 의심해볼 수는 있음, 선형 관계는 0이지만 비선형 관계는 살펴봐야함)

**통계 : 임의추출에서 뽑는 표본들이 독립&동일분포 이어야 통계의 가정들이 만족됨**

임의 추출이란?

* iid : 독립, (모집단과)동일분포
* 독립이기 위해서는 모집단의 크기가 매우 크면 됨 or 표본에 비해 매우 크면 됨

numpy는 데이터를 기본적으로 샘플이라고 생각하지 않음 ⇒ ddof 옵션을 지정해줘야함

Default는 var 구할 시 n으로 나눔

pandas는 데이터를 기본적으로 샘플이라고 생각하기 때문에 var 구할시 n-1로 나눔