0. 강사 및 강의 소개

강사 소개

기본 사항

김민준 | 1993년 11월 22일 | 대구 출생 | 이대역 근처 거주

서울대학교 자유전공학부 오랜 기간 동안 휴학 중 | 사회복무요원 복무 중

010-5511-4898 문자 전화 카톡 환영 | lakiu@naver.com

이력 및 수상

대구 대륜고등학교 졸업 | 수리 가형 + 사회탐구 응시

EBS 공부의왕도 출연 | 청소년을 위한 만만한 경제학 저술

TESAT 최우수상 | 매경TEST 대상 | 경제학원론(조순 外) 10판 개정작업 참여

삼성전자 C-Lab 인턴십 - VR 관련 기술 프로젝트 참여

인디게임 Bytes of Hexagon 기획 & 개발

색 조합 추천 딥러닝 "Daltonism" 개발

영어교육 웹 솔루션 스타트업 "Flowenglish" CTO

용산구 전공연구반 (2013년~) | 성심여고 방과후학교 (2012년~)

세종시 전공연구반 (2017년~)

관심과 취향

딥러닝, 데이터과학, 게임제작, 경제학, 창업, 특이점, 인공지능, VR, 3D 프린터

책 모으기, 게임하기, 오늘 할 일 미루기, 저축 안 하고 이거저거 막 지르기

호: 고래, 모찌, 부대찌개 | 불호: 카카오프렌즈

강의 목표

데이터과학과 머신러닝을 이해하고, Azure 클라우드 서비스를 통해 실습한다.

경영 혁신 및 사회적 가치 창출을 위한 데이터와 인공지능의 활용을 분석한다.

데이터 분석에 필요한 통계학 이론과 개념을 이해하고 실습을 통해 적용한다.

강의 구성

통계학 & 데이터과학 & 머신러닝

기본적인 통계 이론과 개념을 공부한 뒤, 이를 데이터에 적용해보는 것을 통해 데이터에서 의미와 가치를 만드는 과정을 실습하는 것을 목표로 한다. 나아가 이러한 데이터를 딥러닝에 활용하는 방안에 대해 탐구한다.

통계학은 고등학교 수준의 내용부터 대학교 1학년 과정의 내용을 다루며, 이는 『그림으로 설명하는 개념 쏙쏙 통계학』과 edX의 "DAT101x: Data Science Orientation" 강의를 참조하였다.

데이터마이닝과 머신러닝은 Azure 클라우드 서비스를 이용할 것이며, 어려운 코드를 작성하지 않고 배운 개념과 기법을 빠르고 강력하게 구현할 수 있다.

기업과 사회에서의 활용

데이터와 통계를 활용한 사례를 살펴보는 것을 통해 상황에 따른 적절한 데이터 활용 시나리오를 생각하고 설계한다. 다양한 실제 사례와 실습을 위한 모의 데이터를 함께 사용할 것이며, 수업 중 배운 내용을 직접 적용하는 것을 통해 특성과 장단점, 한계를 알 수 있도록 하였다

수강생들은 강의 진행 기간 동안 자신만의 데이터 활용 시나리오를 기획하고, 데이터를 수집한 뒤 분석 결과를 발표해야 한다.

일정

데이터, 통계학, 데이터 과학 / 머신러닝

데이터를 활용하는 방법

통계학 기초 이론

데이터의 수집과 가공

정규분포 | 통계적 검정

회귀분석

확률 | 베이즈 정리

분류 | ROC Curve

군집분석

데이터 시각화

머신러닝 서비스 배포

딥러닝의 구조와 학습과정

데이터 활용 시나리오 프로젝트 발표

유의사항

강의의 목적 및 한계

본 강의는 짧은 일정으로 인해, 데이터과학 및 통계학의 이론적 측면을 세세하게 탐구하는 대신 실습과 활용에 필요한 부분만을 다룰 것이다. 이에 대해 추가적으로 심화학습이 필요한 부분은 앞서 언급한 『그림으로 설명하는 개념 쏙쏙 통계학』과 edX의 "DAT101x: Data Science Orientation" 강의와 함께 『Head First Statistics』와 edX의 다른 강의들을 통해 보충할 수 있을 것이다.

사례

최대한 많은 분야에 적용될 수 있고, 주변에서 접할 수 있는 사례를 활용할 것이나, 만약 학생이 관심있는 사례나 분야가 있다면 반영할 것이다.

실습

매 강의 이후 집에서 Office Mix를 통해 Lab을 수행하고 퀴즈를 풀 것이다. 핵심 개념을 복습하고, 엑셀 실습 영상을 따라한 뒤 그 결과를 묻는 퀴즈를 해결한다.

데이터 분석 시나리오

관심있는 분야에서 데이터를 활용하고 분석하는 시나리오를 기획하고, 실제로 데이터를 수집한 뒤 강의에서 다룬 기법을 적용한 뒤 발표한다. **시나리오 발표와 Lab을 모두 수행해야만 강의를 수료할 수 있다.**

Azure Machine Learning Studio

Microsoft가 제공하는 클라우드 기반 머신러닝 서비스로, 데이터마이닝 및 머신러닝을 코드를 작성하거나 복잡한 설치과정 없이 간단하게 웹에서 무료로 수행할 수 있다.

https://studio.azureml.net/

오픈 채팅방

https://open.kakao.com/o/g5KKEZv

강의자료 및 공지사항, 질문답변 등

1. 데이터, 통계학, 데이터과학

#데이터 #데이터의종류 #빅데이터 #3V
#통계학 #데이터과학 #데이터활용시나리오

데이터란?

Data라는 단어는 라틴어 Datum의 복수형인 Data에서 유래했으며, 그 뜻은 재료, 자료, 논거 등이 있다. 우리 말로 하자면 자료라고 할 수 있겠으나 그 미묘한 어감이 사라지는 관계로 그냥 다들 데이터라고 한다. 발음은 사실 데이ㅌㅏ 정도가 맞지만 이미 표준어로 굳어졌다. 최근 빅데이터 열풍을 타고 마케팅에서 마법의 단어로 사용되고 있으며, 뭔가 체계적이고, 객관적이고, 신뢰할 수 있고, 비싼 값을 받아도 될 거 같은 이미지를 형성하기 위해 쓰인다. 때로는 스마트폰, 인터넷 요금제에서 데이터 사용량 내지는 제공량을 데이터라고도 한다.

데이터를 정확하게 정의하기는 어렵지만, 어떤 정보를 가진 값, 숫자, 문자, 영상, 그림, 소리 등을 모두 데이터라고 할 수 있다. 21세기에는 이 모든 것을 결국 컴퓨터 파일로 저장하고 처리하기 때문에, 데이터는 곧 파일이고 파일이 곧 데이터가 된다고 봐도 큰 문제는 없을 것이다.

데이터의 종류

데이터는 다양한 기준을 통해 종류를 구분할 수 있으며, 다른 종류의 데이터는 다른 방식으로 처리되어야 한다. 따라서 이를 파악하는 것은 매우 중요하다.

하나의 데이터는 다양한 기준들에 대해 여러 종류로 구분될 수 있다. 가령 나이는 양적 | 이산 | 정형 데이터에 속한다.

질적 데이터 VS 양적 데이터

질적 데이터는 남자, 여자와 같은 범주(Category) 자료와 등급, 순위와 같은 순서(Ordinal) 자료가 포함된다. 이들에게 임의의 숫자를 부여할 수도 있으나 (남자는 0, 여자는 1 등) 이 숫자를 평균을 내거나 더하는 등의 연산은 무의미하다.

양적 데이터에는 온도와 같이 수치의 간격(Interval)이 의미를 가지는 자료와 간격과 함께 비율 (Ratio)도 의미를 가지는 무게, 시간 등의 자료가 있다.

연속 데이터 VS 이산 데이터

길이, 면적, 시간 등 연속성이 있는 자료를 연속 데이터라 하고, 주사위 눈, 등급, 나이 등 연속성이 없는 자료를 이산 데이터라고 한다. 간단히 말해 실수로 표현되어야 하는 것은 연속, 정수로 표현되는 것은 이산이라고 볼 수 있다.

정형 데이터 VS 비정형 데이터

형식이 정해진 데이터, 즉 신체검사 기록, 연락처, 주문 기록 등 표로 나타낼 수 있는 데이터를 정형 데이터라고 한다. 전통적으로 데이터의 분석과 수집 모두 정형 데이터 위주로 이루어졌으나, 최근에는 이미지, 비디오 등 다양한 비정형 데이터의 중요성이 높아지고 있다.

빅데이터

최근 자주 사용되는 빅데이터라는 단어는 그만큼 자주 오용되기도 한다. 대개의 경우 마케팅 용어로 사용하는 빅데이터는 빅데이터가 아닌 경우가 많은데, 이는 빅데이터를 정의하는 3V를 만족하지 못하기 때문이다.

Volume

빅데이터는 이름에서 알 수 있듯 양이 많다. 이 양의 기준은 절대적인 것이 아니라, 데이터를 분석하고 활용하는 목적에 따라 달라진다.

Velocity

데이터 입출력에 필요한 속도를 의미한다. 양이 많더라도 기존의 기술로 필요한 입출력을 감당할 수 있다면 빅데이터로 보기 힘들다.

Variety

다루는 데이터의 종류가 다양함을 의미한다. 비정형 데이터의 처리가 빅데이터의 핵심인 이유기도 하다.

통계학

통계학은 데이터를 관찰하고 처리하는 것을 연구하는 학문이라고 할 수 있다. 데이터의 수집, 가공, 분석, 처리, 의미 추출 등이 통계학의 범위에 포함된다. 그 특성상 굉장히 실증적인 성향을 보이며, 동시에 이론적 토대도 굳건한 학문이다. 다시 말해 어렵다. 문제는 이 통계학이 쓰이지 않는 곳이 없다는 것이다.

현 시대는 정보가 모든 것을 지배하고 있다. 모든 기업은 나름의 방식으로 정보를 수집하고 가공하여 의사결정을 한다는 점에서 IT 기업이다. 개개인의 의사결정에도 정보가 중요한 것은 말할 것도 없다. 이 정보를 처리하는 방법이 바로 통계라는 점에서 통계학은 이 시대를 살아가는 모든 이들의 필수 학문이 되었다.

순수 인문대를 제외한 다른 모든 전공에는 통계학 과목이 존재한다. 대학 문턱을 밟는 순간 여러분은 통계학의 세계에 강제로 입장하게 되는 셈이다. 순수 인문대의 현실을 생각해보면 결국 먹고 살려면 통계를 해야한다는 결론이 나온다.

한편 통계는 이 정보를 정확하게 처리하여 올바른 선택을 내리는 것이 목적이지만, 반대로 정보를 왜곡하여 잘못된 선택을 유도하도록 악용되기도 한다. 이러한 사기에 당하지 않기 위해서도 통계학을 배울 필요가 있다.

데이터과학

데이터과학은 근래에 급부상한 단어로, 통일된 정의는 없으나 프로그래밍, 통계학, 수학, 기계학습, 그리고 각 분야에 대한 지식(Domain knowledge)가 데이터과학자가 갖춰야 할 소양이라고 볼 수 있다. 본래 통계학이 데이터를 분석하는 학문이란 점에서 겹치기도 하나, 여기에 프로그래밍과 배경지식이 추가되어 실제 기업과 사회 현장에서 데이터를 활용하는 시나리오를 계획하고, 이에 필요한 솔루션을 만들고 실행해 의사결정에 영향을 주는 일련의 과정이 중심이라는 점에서 구분할 수 있다.

데이터과학자는 현재 인기도 1위의 직업으로, 평균 연봉은 $116,840 약 1.35억이며, 데이터과학자에 대한 수요는 날이 갈수록 급증하고 있다.

데이터의 수집과 활용 시나리오

목표 설정

데이터를 수집하고 활용하려는 목적을 명확하게 설정한다. 사용 가능한 데이터의 종류와 양을 기반으로 당면한 문제에 도움을 줄 수 있는 솔루션을 찾는다.

수집

데이터를 수집한다. 내부 데이터, 웹 크롤링, API, 센서, 공공 데이터 등 이미 존재하는 데이터를 이용하거나 새로운 데이터를 수집한다.

저장

데이터를 저장한다. 데이터의 유형과 양에 따라 저장 자체가 개인 수준에서는 어려울 수 있으며, 이 경우에는 전용의 스토리지 서버를 임대하거나 구축할 필요가 있다. 또한 데이터의 백업과 보안도 신경 써야 할 요소다. 최근에는 클라우드 서비스를 통해 저비용으로 고성능의 서비스를 이용할 수 있다.

가공

데이터를 목적에 맞게 가공한다. 여러 출처에서 데이터를 수집했을 경우, 양식 등을 통일 시킬 필요가 있다. 또한 누락, 중복 등의 문제를 해결해야 한다. 필요한 경우 이 과정에서 기계학습을 통해 데이터를 1차적으로 가공할 수도 있다.

쿼리 및 마이닝

특정 조건을 만족하는 데이터를 찾거나, 데이터의 평균값이나 분산 등의 변수를 계산하는 등의 작업을 수행한다. 각종 통계적 기법과 기계학습을 통해 데이터에서 의미를 추출한다.

실제 활용

데이터에서 추출한 의미나 정보를 시각화 혹은 보고서 등으로 작성하고, 팀 및 경영진에게 데이터가 주는 메시지를 효과적으로 전달한다.

2. 머신러닝, 딥러닝

#머신러닝 #데이터마이닝 #딥러닝
#ANN #CNN #RNN #강화학습 #GPGPU

머신러닝이란?

기존의 프로그램과 달리, 컴퓨터가 스스로 주어진 문제로부터 학습하고 주어진 데이터에 대한 평가와 새롭게 들어올 데이터를 처리할 수 있는 능력을 갖추게 하는 것을 의미한다.

데이터에서 의미 있는 규칙과 패턴을 찾는 데이터마이닝과 비슷해 보일 수도 있고, 실제로 기계학습의 일부 과정에 데이터마이닝이 쓰이기도 하나 기계학습은 데이터의 검증과 예측이 목표란 점이 다르다.

머신러닝이 각광받는 이유

데이터는 머신러닝을 위한 자원이다. 기계학습의 본질은 결국 데이터의 분류와 검증, 예측이기 때문이다. 좋은 데이터가 많을수록 심화된 학습이 가능하며, 이로부터 더 정확하고 세밀한 처리 능력이 개발된다.

한편, IT혁명 이후로 웹, 스마트폰, SNS, 사물인터넷의 발달과 확산은 무수한 데이터의 생산과 수집이 가능해졌다. 본래 이러한 데이터는 정리나 분류가 되어있지 않아 쓰레기에 가까웠으나 데이터마이닝, 빅데이터 기술의 발달로 데이터에서 의미 있는 정보를 빠르게 추출할 수 있게 됐다. 이용할 수 있는 데이터의 양과 질의 획기적 개선과 GPGPU를 비롯한 컴퓨터 성능의 발전은 머신러닝의 잠재력을 폭발시켰으며, 수많은 기업과 사회가 머신러닝에 주목하는 이유가 됐다.

딥러닝이란?

기계학습의 한 분야로, 사람의 뇌와 같은 뉴런 신경망을 인공적으로 구현한 ANN(Artificial Neural Network)를 기반으로 한 알고리즘들을 통칭한다. 다른 방법과 달리 높은 추상화가 특징이며, 사람이 직접 규칙을 지정하지 않아도 네트워크 스스로 문제에 대한 해결법을 구축하는 듯한 모습이 장점이자, 그 내부 구조를 개발자도 정확히 파악하기 어렵다는 점에서 단점이 되기도 한다.

딥러닝이 각광받는 이유

딥러닝의 원류인 인공신경망은 학계에서 사장됐던 분야이다. 학습이 매우 느리고 부정확했으며, 사전에 규칙을 넣어주지 않아도 되는 만큼 더 많은 데이터가 필요했기 때문이다. 그러나 학습 알고리즘의 개선과 함께 컴퓨터의 발달, 특히 GPU의 발달로 인해 학습에 걸리는 시간이 획기적으로 줄어들었으며, 빅데이터의 등장은 바로 이러한 딥러닝에 필요한 무지막지한 양의 데이터를 마련할 수 있게 하였다. 내외적으로 모든 조건이 딥러닝에게 최고의 환경을 조성한 것이다.

또한 딥러닝을 위한 여러 기법이 개발되고 연구되며 딥러닝의 잠재력 또한 더 커졌다. 초기 딥러닝은 숫자 몇 개를 입력으로 받아 역시 숫자로 된 출력을 내놓는 정도였으나, CNN(Convolution Neural Network)를 활용한 이미지 분석, RNN(Recurrent Neural Network)를 활용한 시계열, 자연어 분석 등을 통해 더 많은 데이터와 문제에 딥러닝을 적용할 수 있게 됐다. 나아가 알파고와 같은 강화학습(Reinforcement Learning)은 게임이나 운전과 같은 분야에 딥러닝을 사용할 수 있는 가능성을 열었다.

GPGPU

General Purpose computing on Graphics Processing Units의 약자로, 일반적으로 2D, 3D 이미지를 처리하기 위한 장치인 GPU를 수치연산 등의 다른 프로그램의 계산에 활용하는 것을 말한다. CPU는 소수의 천재가 있는 방이라면, GPU는 수천의 평범이들이 있는 방이라 볼 수 있으며, 단순하고 병렬화하기 쉬운 연산에 대해선 CPU 대비 수 백, 수 천 배의 속도를 낼 수 있다.

간단한 머신러닝의 경우 일반적인 CPU로도 충분하나, 딥러닝의 경우 매우 간단한 경우를 제외하면 GPU의 사용이 필수적이다. 대부분 NVIDA의 그래픽카드를 사용하며, Tesla, Titan X(p), GTX 1080 (Ti) 등 최고사양의 제품이 딥러닝 연구, 개발을 위한 컴퓨터에 장착되고 있다. 또한 AWS, Azure 등 클라우드 서비스에서도 GPU가 장착된 서버를 합리적인 가격에 대여할 수 있다.

Race against the Machine

데이터과학과 머신러닝, 딥러닝은 막대한 가치를 창출하며, 과거에는 수많은 인력과 비용이 들었을 일을 빠르게 자동으로 처리하게 한다. 이 말을 뒤집으면 어떻게 되는가? 기업은 비용절감을 위해 기계가 데이터를 수집하고 처리하는 것보다 느린 노동자를 해고할 것이다. 실제로 최근 기술 발달로 인해 사라질 일자리들에 대한 예측이 쏟아져 나오고 있다.

과거에는 타이피스트란 직업이 있었다. 타자가 일인 직업이다. 각 부서에서 펜으로 쓴 메모나 회의록을 가져오면 타자기로 쳐서 문서로 만드는 일을 했다. 지금 타이피스트는 사라졌다. 타자를 대신 쳐준다는 것도 웃기지만 애당초 타자를 칠 필요가 사라지고 있다. 고객상담센터에 전화를 했는데, ARS가 응대를 하고 음성인식을 통해 담당자를 연결해 준 경험이 있는가? 담당자는 살아남았지만 응대를 하던 사람의 일자리는 사라진 것이다.

한편 오토 파일럿이 등장했지만 조종사는 사라지지 않았다. 사라진 것은 항법사, 항공기관사다. 로봇 변호사 ROSS가 미국의 대형 법무법인에 채용됐다. ROSS는 변호사들과 함께 일한다. 변호사들과 함께 판례를 검색하고 분석하던 사무원들은 사라졌다. 이제 우리는 기계가 대체할 직업과, 대체하지 못할 직업에 대해 생각할 필요가 있다. 그 차이는 무엇인가?

감성이나 감정이 필요한 일자리는 살아남을 것이다. 의사결정권을 가진 직업도 살아남을 것이다. 그러나 그 일을 보조하는 일자리는 크게 줄어들 것이다. 다시 말해 데이터를 처리하고 분석하는 직업은 전망이 좋지 않다. 업무의 매뉴얼이 구체적이고 반복적일수록 기계에 쉽게 대체될 것이다. 텔레마케터, 상담원, 운전기사가 그 예가 될 것이다.

Q. **기계와의 경주에서 유리한 / 불리한 직업**?

Race with the Machine

기계가 인간보다 나은 분야는 분명 많으나, 인간을 완전히 대체할 수 있는 분야는 한정되어 있다. 반면 인간이 기계를 통해 생산성을 역사상 전례없이 끌어올릴 수 있는 기회는 많다. 인간이 수행하기에 복잡하거나, 귀찮거나 시간이 오래 걸릴 작업은 기계에게 맡기고 중요한 작업에 집중할 수 있는 것이다.

명함을 예로 들어보자, 직장인들에게 명함은 중요한 비즈니스 요소다. 상대의 소속, 직급, 이름, 연락처를 파악해야만 결례를 범하지 않으며, 필요한 일이 있을 때 즉각 연락할 수 있다. 그러나 명함을 받고 일일이 기록하는 것은 번거롭다. 명함 인식 앱은 그저 명함을 카메라로 찍기만 하면 이러한 정보를 자동으로 인식하고, 연락처에 추가해준다.

또한 기계를 이용하여 새로운 가치를 창출할 수도 있다. SNS는 끝없이 수많은 사람들의 의견이 자유롭게 게시된다는 점에서 빅데이터와 기계학습의 주목을 받고 있다. 특히, 트렌드 분석에서 그 잠재력과 성과를 입증하고 있다. 특정인, 특정 단체에 대한 글에서 호와 불호를 판별하고, 이들의 평소 행동과 정보를 결합해 분석하는 것이다.

Deus ex machina

딥러닝의 특징은 우리가 원리나 규칙, 알고리즘을 모르는 일에 대해서도 기계가 스스로 학습하여 처리할 수 있다는 것이다. 지금까지의 기계는 설계자의 능력에 제한되어 왔으나, 이제는 그 제한이 사라지는 것이다. 이는 극단적으로 말해, 충분히 많은 양의 데이터와 충분히 빠른 컴퓨터만 있으면 인간보다 뛰어난 기계를 만들 수 있다는 것이다.

실제로 사진에 있는 항목을 분류하는 ImageNet 경진대회에서의 우승 프로그램들은 이미 인간보다 더 우수한 성적을 보이고 있다. 알파고는 이미 인간과 비교할 수 없는 수준의 실력을 갖췄으며, 포커대회도 인공지능이 승리하였다. 문장의 구문을 분석하는 구글의 클라우드 서비스는 숙련된 모국어 화자의 정확도보다 더 높은 정확도를 보이고 있다.

3. 데이터를 활용하는 방법

#존스노우 #심야버스 #TARGET #구글어스
#데이터분석도구 #데이터리터러시

데이터 분석 사례

존 스노우

왕좌의 게임 등장인물이 아니라, 빅토리아 시대의 의사로 본래는 마취방법 등을 연구하며 빅토리아 여왕을 직접 치료하기도 하였다. 아이러니하게도 후대에는 마취가 아닌 역학조사의 선구자로서 유명하다. 현대의 역학은 최첨단 통계기법과 빅데이터를 통해 발전했지만, 그 근본은 존 스노우의 콜레라 조사에 있다.

당시 런던의 상하수도는 엉망진창에 가까웠으며, 따라서 콜레라와 같은 수인성 전염병이 창궐하였다. 스노우는 콜레라 환자의 집을 지도에 표시해보았고, 거리의 특정한 지점을 중심으로 콜레라가 점점 퍼진다는 사실을 찾아낸다. 그 지점은 바로 펌프였다.

허나 당대 의학계에서는 전염병의 원인이 물이 아닌 미아즈마라는 공기 중의 독기라는 의견이 절대적 지지를 받고있었으며, 펌프의 물을 퍼올려 현미경으로 보아도 특별한 것을 찾을 수 없었다. 스노우는 이에 좌절하지 않고 환자들이 실제로 펌프의 물을 마셨는지 집요하게 확인하고, 먼 지역에서 발병한 환자가 사실은 그 펌프물을 가져와 마셨다는 사실 등을 밝혀내며 기어이 지역사회를 설득해 펌프를 폐쇄하고 콜레라를 종식시킨다. 이후 해당 펌프 근처에서 원인요소를 찾은 결과 첫 발병자의 집 정화조가 펌프와 지나치게 가까이 있었고, 부식된 벽으로 인해 오염이 발생한 것이다.

심야버스

서울시가 처음 심야버스를 계획하면서 한정된 버스 노선으로 최대한 많은 지역과 이용자를 커버해야 한다는 문제에 직면했다. 또한 심야 시간대에는 낮과는 다른 이용 패턴이 나타나기에, 문제는 더욱 복잡해졌다.

이는 서울시가 가진 택시 승하차 데이터를 분석하는 것을 통해 해결됐다. 심야에 가장 승하차가 많은 구간을 중심으로 노선을 계획한 것이다.

TARGET

타겟은 미국내 업계 2위의 할인매장으로, 2002년 고객 데이터 분석 서비스 부서를 신설한 이후로 빅데이터 분석을 통해 고객들에게 맞춤 쿠폰을 보내고 있다. 과거의 구매 내역을 추적해 자주 사는 품목의 신제품 및 경쟁제품에 대한 쿠폰을 보내거나 내역에 변동이 있을 경우 이를 예측해 곧 구매하게 될 물건에 대한 할인을 제공하는 것이다.

타겟의 맞춤 쿠폰은 2012년 10대 여고생에게 유아용품 할인 쿠폰을 보낸 사건이 알려지면서 유명세를 탔다. 쿠폰을 본 아버지는 딸이 임신이라도 한 줄 아냐며 타겟에 강력하게 항의했으나, 상황은 며칠만에 반전된다. 그 학생이 정말로 임신을 한 것이었기 때문이다.

타겟은 임산부의 일반적인 구매 패턴을 통해 특정 고객의 임신 여부를 판단한다. 가령 임신 초기에는 칼슘, 마그네슘 등이 포함된 영양제를 구매하고, 20주를 넘으면 튼살 방지를 위한 로션을, 출산이 임박하면 유아용품을 구매한다는 것이다. 학생이 영양제와 로션을 일정 기간을 두고 구매한 것이 포착되었고, 이를 역산하여 출산시기를 계산해 때에 맞춰 쿠폰을 보낸 것이다.

동물들은 지구 자기장을 느낀다

사빈 베갈이 이끄는 연구팀은 동물들의 자기 감각(Magnetic sense), 즉 지구 자기장을 감지할 수 있는 능력에 대해 연구하고 있었다. 새나 물고기, 쥐 등이 이 감각을 가지고 있다는 것은 많이 알려졌으나, 문제는 더 큰 동물에게도 같은 능력이 있냐는 것이었다. 이들은 구글 어스를 이용해 위성사진에 찍힌 소와 사슴들의 자세를 조사하였고, 머리가 북쪽을 향하려는 경향이 있음을 밝혀냈다.

Q. **사례들의 공통점과 배울 점은 무엇인가?**

데이터를 처리하기 위한 도구

EXCEL

종종 과소평가 당하기도 하나, 일반적인 개인용 컴퓨터로는 감당할 수 없는 양의 데이터를 처리하거나 매우 복잡한 연산을 하는 것이 아니라면 엑셀의 기능적 한계로 분석을 못할 경우는 거의 없다. 또한 필요한 경우 VBA나 각종 추가기능을 통해 전문 솔루션 못지않는 효율성을 보일 수도 있다.

Python

파이썬은 배우기 쉽지만 강력한 프로그래밍 언어로, 일반적인 코딩부터 웹, 서버, 데이터 마이닝, 인공지능 등에 이르기까지 매우 다양한 영역에서 사용되고 있다. 파이썬은 웹 크롤링이나 데이터 전처리 등을 적은 비용을 통해 할 수 있다는 점에서 그 자체로도 뛰어나지만, 언어와 솔루션을 가리지 않고 궁합이 잘 맞는다는 점에서 이들 사이의 다리 역할을 한다는 것도 빼놓을 수 없는 장점이다.

“Life is too short. You need python.”

R, Matlab

파이썬을 비롯한 범용 프로그래밍 언어와 달리, 이들은 태생이 수치 연산과 통계 처리에 있다. 특히 기업들이 주로 파이썬을 이용하는 경우가 많은 반면, 학계에서는 R과 매트랩을 주로 사용하는 편이다. 많은 연구, 활용 사례와 확장기능 등이 있으며 기본적으로 데이터에 최적화되어 있는 것이 장점이다.

데이터베이스 관리 시스템 - SQL, 하둡 등

SQL은 데이터베이스를 관리하고 질의하기 위한 프로그래밍 언어로, 웹페이지, 재고관리, 각종 프로그램 등 다양한 곳에서 표준으로 사용되고 있다. 따라서 SQL은 데이터 처리의 표준어라고도 볼 수 있다.

반면 빅데이터의 경우는 SQL로는 한계가 있기에, 클라우드나 그리드 컴퓨팅을 통한 분산처리를 지원하는 하둡 등의 빅데이터 프레임워크가 이용되고 있으며, 구글, 페이스북을 비롯한 IT 공룡들은 자체적으로 솔루션을 구축하고 있다.

데이터 분석을 위해서는

기업, 조직에서 데이터 분석을 의사결정에 활용하는 것은 단순한 문제가 아니다. 데이터 분석 담당자만이 아니라 구성원 모두의 협조와 결정권자의 열린 태도가 필요하며, 데이터의 품질 유지와 관리를 위한 밑작업과 업무 프로세스 개선도 동반되어야 한다. 그러나 데이터 분석에 대한 수요는 모든 산업에서 급속도로 늘어나고 있으며, 보유한 데이터에서 가치를 창출하는 것이 경쟁에서의 핵심이 되면서 이를 위한 준비가 되지 않은 기업은 곧 도태될 전망이다.

또한 최근에는 Quantified self, 즉 개인의 일상생활에서 자신의 신체 상태를 각종 스마트, 웨어러블 기기로 측정하고 관리하는 것을 통해 삶의 질을 높이려는 움직임이 확산되고 있다. 체중, 심박수, 걸음, 운동 기록, 수면, 영양 정보 등의 수많은 생활 데이터가 다양한 센서를 통해 수집되고 있으며, 한 연구결과에 따르면 미국 성인의 69%가 적어도 하나의 생활 데이터를 추적하고 있다.

이렇듯 데이터 분석은 모두에게나 중요하지만, 그만큼 준비해야할 것도 많다.

기록하기

분석을 하기위해선 당연히 데이터가 존재해야 한다. 데이터 자체가 파일 등으로 잘 관리되고, 백업되고 있을 뿐만 아니라 데이터의 생성 및 수집 과정에 관한 메타데이터도 착실히 있어야 한다. 공장이나 사무실 등의 시설, 차량 등에 센서를 부착하고 네트워크에 연결하는 것도 떠오르는 경영 혁신 사례다.

Data literacy

글을 읽고 해석하며 맥락을 파악하는 것처럼, 데이터 리터러시는 데이터를 읽고 해석하며 의미를 파악하는 능력을 말한다.

열린 마음

모든 변화와 혁신이 요구하는 것이지만, 기존의 프로세스를 고수하는 대신 열린 마음으로 새로운 아이디어와 방법을 받아들일 필요가 있다. 당장은 조금 귀찮더라도, 데이터를 착실히 기록한다면 이는 곧 커다란 자산이 되어 돌아올 것이다.

4. 데이터의 수집과 가공

#데이터베이스 #크롤링 #API #전처리
#회귀분석 #군집분석

데이터 수집

데이터베이스

리빙포인트: 데이터를 수집하려면 데이터가 저장되어 있는 데이터베이스를 찾아 연결하면 된다. 문제는 세상 어디에도 "내가 필요한 데이터가 저장된 DB" 같은 건 없다는 것이다. 설사 존재한다 하더라도 외부에 공개되어 있지 않거나, 불완전한 경우가 많다. 이를 극복하기 위해서 문제에 관련된 데이터를 담고 있는 DB가 무엇이 있는지 탐색하고, 때로는 심야버스 노선을 위해서 택시 데이터를 이용한 것처럼 창의력과 상상력을 발휘해야 할 수도 있다.

문헌자료

논문이나 보고서의 표, 차트, 각종 문서도 전통적이지만 여전히 유용한 데이터 소스다. 디지털화가 되어 있지 않다면 파일로 만드는 과정이 수반되어야 한다는 단점이 있으나, 때로는 유일한 대안인 경우도 있다.

웹 크롤링

웹 페이지와 게시판 등의 자료를 문자 그대로 긁어온다. 해당 사이트의 구조와 HTML에 대한 이해가 있어야 원하는 데이터를 추출할 수 있다. 주로 파이썬과 크롬 개발자 도구를 이용해 웹 크롤러를 제작한다.

API (Application Programming Interface)

API란 특정 서비스나 소프트웨어에 대해 외부에서 접근할 수 있도록 만들어 둔 기능이라고 할 수 있다. 가령 페이스북 API는 다른 프로그램이 페이스북 계정에 대한 정보를 알아내고 연동이 가능하게 한다.

Digital Exhaust

페이지 방문기록, 쿠키, 검색, 머무른 시간 등 인터넷을 사용하면서 자연스럽게 생기는 정보들을 의미한다. 구글 맞춤 광고 등이 이를 사용하는 대표적 사례다.

데이터 전처리

변수 확인 및 형태 변형

데이터 처리에 앞서 해당 데이터의 메타데이터를 파악한다. 즉 필드와 레코드의 의미, 출처 및 수집 과정에 대해 확인한다. 몇몇 데이터는 분석 기법을 적용하기 어려운 형태로 저장되어 있을 수 있는데, 이 또한 적절하게 변형한다.

단위 및 양식 통일

다양한 소스에서 데이터를 수집한 경우, 같은 변량에 대해서도 다른 단위나 양식을 사용할 수 있다. 섭씨, 화씨, 미터, 야드 등이 대표적이며 이외에도 주소, 전화번호 등도 형식을 통일시킬 필요가 있다.

중복 및 누락, 이상값

동일한 레코드가 여러 개 있다면 하나만 남기고 제거한다. 단, 이 때 일치성의 판단 기준이 문제가 될 수 있다. 가령 이름은 동명이인이 흔히 존재할 수 있어 적합하지 않다. 누락의 경우 데이터의 특성에 따라 적절한 처리방법을 결정해야 한다. 일반적으로 0으로 놔둘 경우 분석에 심각한 왜곡을 주기 때문에, 집단의 평균을 넣어주거나 앞뒤 변수의 평균을 이용한다. 혹은 참고할 수 있는 다른 값을 이용해도 좋다.

통합

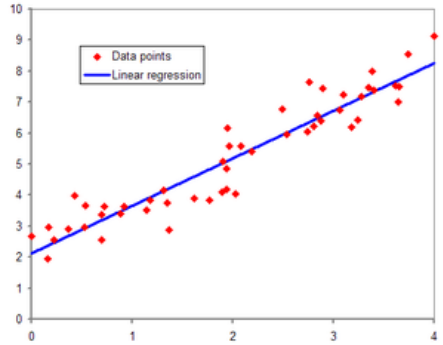
데이터 사이의 관계를 이용해 데이터들을 통합할 수 있다. 대표적으로 주문내역으로, 주문내역에 있는 상품 필드를 재고관리내역의 상품과 연결할 수 있다.

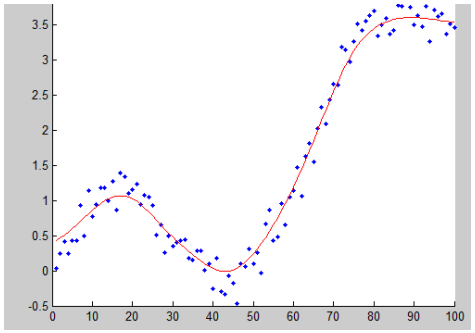
변환

지수나 제곱, 로그적 분포를 보이는 데이터는 선형으로 변환하면 더 좋은 결과를 얻을 수 있다. 또한 텍스트나 음성, 영상 등의 비정형 데이터는 문장 부호나 노이즈를 제거하고, Word embedding이나 FFT를 통해 변환하는 과정이 필요하다.

회귀분석(Regression Analysis)

회귀분석을 통해 연속 데이터들 사이의 관계를 설명하는 모형을 구할 수 있다. 아래와 같은 선형회귀가 대표적이며, 간단하게는 변수들의 분포를 가장 그럴듯하게 보여주는 선이나 함수를 찾아내는 과정이라고 이해할 수 있다.





회귀분석을 통해 경향성을 찾아낼 수 있으며, 이러한 경향이 데이터를 얼마나 설명할 수 있는지를 적합도라는 지표를 통해 알아낼 수도 있다. 또한 회귀분석을 통해 독립변수와 종속변수의 관계를 수치적으로 알아낼 수 있다. 전단지를 돌리는 것이 매출에 어떤 영향을 미치는가? 100장을 돌리면 얼마를 더 버는가? 와 같은 질문에 대해 답할 수 있는 것이다. 또한 데이터에 없는 경우에 대해서도 예측을 할 수 있다.

선형회귀의 경우, 데이터의 관계를 아래와 같은 식으로 나타낸다.

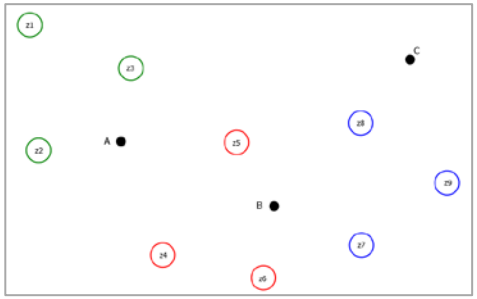
$$Y = aX_1 + bX_2 + cX_3 + \cdots + z$$

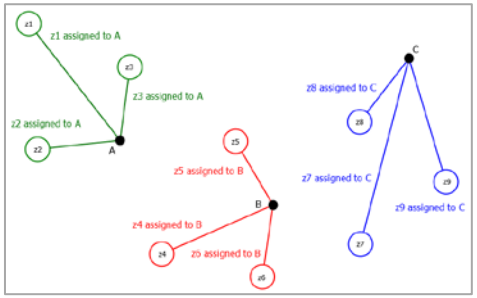
각 X들 앞에 있는 계수는 곧 그 변수가 Y에 미치는 영향이며, z는 절편이 된다. 가령 손님 수 = 0.1 * 전단지 수 + 10이라면 전단지를 10장 더 돌리면 손님이 1명 더 올 것이라는 예측을 할 수 있는 것이다. 이렇듯 선형회귀는 해석이 간단하기에 유용하게 사용되나, 변수들 사이에 선형 관계가 없으면 이용할 수 없다. 이 경우 변수에 로그나 제곱, 루트 등을 취해 선형으로 변환하는 방법이 있으며 이를 통해 다양한 상황에서 선형회귀를 적용할 수 있다.

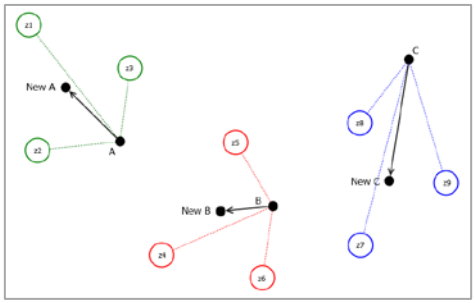
군집 분석(Cluster Analysis)

데이터의 특성을 더 잘 이해하고 구분하기 위해 데이터를 비슷한 것끼리 그룹화하는 것을 군집화라고 한다. 군집 분석은 데이터를 잘 설명할 수 있는 그룹을 정하고, 각 그룹의 대표점을 찾는 데이터 마이닝 기법이다. 마케팅 및 고객관리, 제품분석, 실험 결과 분석 등 다양한 목적을 위해 유용하게 쓰인다.

군집화를 위한 알고리즘은 매우 다양하나, 가장 널리 알려진 것은 k-means 알고리즘이다. 이 알고리즘은 주어진 데이터를 k개의 군집으로 묶으며, 각 데이터와 군집의 중심 사이의 거리를 최소화하는 방식으로 작동한다.







Q. **회귀 및 군집 분석을 적용할 수 있는 사례?**

5. 데이터를 설명하는 통계 - 기술통계학

#통계량 #평균 #중앙값 #편차 #분산 #상관계수 #표 #그래프

통계량

데이터의 표본 하나하나를 살펴보며 특성을 알아볼 수도 있지만, 시간과 비용이 많이 들고 알아낼 수 있는 것도 제한적이다. 기술통계학(Descriptive statistics)은 이처럼 방대한 데이터를 표나 그래프로 알아보기 쉽게 나타내고, 각종 통계량을 계산해 데이터의 특성을 요약하는 학문이다.

고등학교 한 반의 내신 성적표를 데이터라고 할 때, 이 표에 학생 한 줄을 레코드라고 한다. 반이 30명이면 이 표에는 30개 레코드가 있는 것이다. 각 레코드에는 국어, 수학, 영어 등의 항목이 있고 해당하는 점수가 있는데, 이 개개의 점수를 변량이라고 한다.

평균(Mean)

$$\text{평균}\ \overline{x} = \sum x_i = \frac{x_1 + x_2 + x_3 + x_4 + \cdots + x_n}{N}$$

N개의 변량에 대해 평균은 변량을 모두 더한 뒤 그 개수인 N으로 나눈 것으로 구한다. 쉽게 구할 수 있으면서도 데이터의 가장 중요한 특성을 나타낸다는 점에서 널리 쓰이나 이상값(Outlier)에 영향을 많이 받는다는 점에서 한계가 있다.

중앙값(Median)

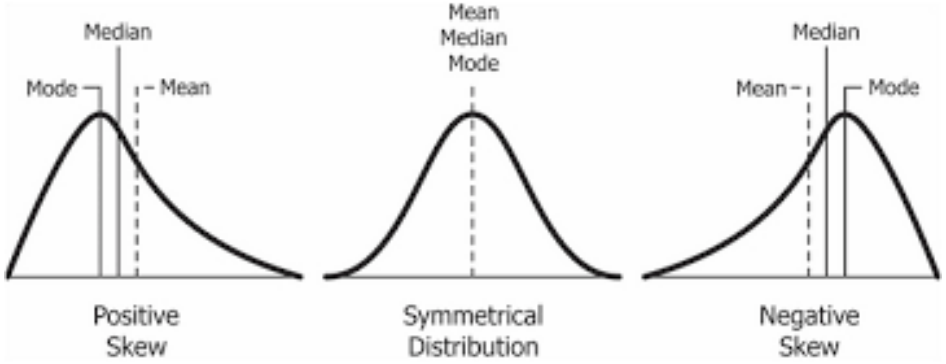
변량을 순서대로 놓았을 때 딱 가운데에 있는 변량의 값을 말한다. 변량이 짝수개이면 가운데의 2개 변량의 값을 평균한 것으로 구한다. 가령 1, 2, 2, 3, 3의 경우 2가 중앙값이며, 1, 2, 2, 3, 3, 3은 2와 3의 평균인 2.5가 중앙값이다.

최빈값(Mode)

변량 중 가장 많이 나타나는 값을 말한다. 1, 1, 1, 1, 2, 2, 2, 3, 3의 경우 1이 가장 많이 나타났으므로 1이 최빈값이 된다.

데이터가 정규분포에 가까울수록 이들 3가지 통계량의 값이 수렴한다.

왜도(Skewness)



데이터의 분포가 치우친 정도를 말한다. 위의 그림에서 볼 수 있듯, 꼬리가 오른쪽으로 길어진 경우 Right tailed라 하며 양의 왜도 값을, 왼쪽으로 길어진 경우 Left tailed라 하며 음의 왜도 값을 가진다. 왜도 값에 따라 평균, 중앙값, 최빈값의 상대적 분포가 변하는 것도 주목하자.

백분위수(Percentile)

변량들을 순서대로 나열했을 때, 백분율로 나타낸 위치에 해당하는 변량의 값을 의미한다. 100개의 변량이 있을 때, 25백분위수의 값은 25번째 변량의 값에 해당한다. 만약 200개의 변량이라면 50번째 변량의 값이 될 것이다. 보통 25, 50, 75 백분위수를 각각 제1, 제2, 제3사분위수라고 한다.

편차(Deviation)와 분산(Variation)

편차는 각 변량에서 평균값을 뺀 것을 의미한다. 이 때, 각 변량들의 편차를 제곱한 뒤 더한 것을 편차제곱합이라고 한다. 편차제곱합을 변량의 수로 나눈 것을 분산이라고 한다. 수식으로는 다음과 같다.

$$V = s^2 = \frac{1}{\mathrm{N}}\{(x_1 - \overline{x})^2 + (x_2 - \overline{x})^2 + \cdots + (x_n - \overline{x})^2\}$$

이 때 분산의 제곱근인 s를 표준편차(Standard deviation)이라고 한다.

분산은 데이터가 평균, 즉 중심으로부터 얼마나 퍼져 있는지를 나타낸다. 이 때 편차를 그냥 더해도 되는데 굳이 저렇게 제곱을 해서 더하는지 의문이 들 수도 있다. 이는 양의 편차와 음의 편차가 서로 상쇄되는 것을 막기 위한 것이다.

가령 2, 2, 2와 1, 2, 3을 생각해보자. 둘다 평균은 2이나 분산은 다르다. 만약 편차를 그냥 더할 경우 전자는 0 + 0 + 0 = 0, 후자는 −1 + 0 + 1 = 0으로 분산이 같은 것으로 나오게 된다. 따라서 이를 막기 위해 편차제곱합을 개수로 나누어 분산을 구한다. 그러나 이렇게 하면서 단위가 바뀌고 (미터에서 제곱미터로 바뀐다거나) 또한 수치가 뻥튀기되기 때문에, 루트를 취해서 다시 원래 단위로 바꾸는 것이다.

어떤 변량의 편차와 표준편차를 비교하는 것을 통해 해당 변량이 평균으로부터 얼마나 떨어져 있는지를 알아볼 수도 있다. IQ가 대표적이다.

공분산(Covariance)

지금까지의 통계량이 한 변수의 특징을 요약했다면, 공분산은 두 변수의 관계를 나타낸다. x와 y의 공분산은 다음과 같이 구한다.

$$S_{xy} = \frac{(x_1 - \overline{x})(y_1 - \overline{y}) + (x_2 - \overline{x})(y_2 - \overline{y}) + \cdots + (x_n - \overline{x})(y_n - \overline{y})}{N}$$

이 값이 양수이면 x와 y가 양의 상관, 즉 x와 y가 같이 증가하거나 감소하고, 음수이면 음의 상관, 즉 x가 증가할 때 y가 감소하거나 그 반대인 관계가 있다. 만약 0인 경우, x와 y 사이에 상관관계가 없다고 본다.

상관계수(Pearson correlation coefficient)

분산과 마찬가지로 공분산도 수치가 단위에 따라 뻥튀기 되는 문제가 있다. 따라서 공분산 값만을 보고 상관관계의 여부나 강도를 알기는 힘들다. 이를 해결하기 위한 것이 아래의 상관계수다.

$$r_{xy} = \frac{S_{xy}}{S_x S_y}$$

이 값은 −1 < r < 1의 범위를 가지며, 1에 가까울수록 양의 상관, −1에 가까울수록 음의 상관을 나타내며 값이 극단적일수록 상관관계가 크다고 볼 수 있다.

표와 그래프

데이터를 분석하고 시각화할 때 표와 그래프를 활용한다. 엑셀을 활용하면 손쉽게 만들 수 있지만, 필요한 정보를 효과적으로 나타내는 것은 또다른 문제다.

표(Table)

표는 행과 열로 구성되어 있다. 엑셀 기준으로 숫자로 표시되는 게 행이며 문자로 표시되는 게 열이다. 하나의 행은 하나의 레코드를 나타낸다. 열은 필드라고도 하며, 각 레코드에 포함된 변량들의 종류를 구분한다. 행과 열이 교차하는 각각의 칸을 셀이라 하며 개별 변량은 이 셀에 들어있다.

표를 활용할 때 가장 중요한 것은 적절한 열을 정하는 것이다. 열은 표의 데이터가 가진 정보의 종류를 정하는 것이기에, 적절한 열을 구성하면 곧 적절한 표가 만들어진다. 표를 막 만드는 대신, 지금 필요한 정보가 무엇인지 고심하고, 필요한 경우 평균, 합계, 순위 등 다양한 통계량을 계산하여 열을 추가한다.

그래프

나이팅게일이 야전에서 병사들의 사망원인을 나타내기 위해 사용한 Rose graph 이후로, 지금까지 수많은 종류의 그래프가 만들어지고 사용되고 있다. 그래프를 사용할 때 가장 중요한 것은 단순히 모양이 이쁜 그래프를 골라서는 안 된다는 것이다. 각 데이터는 그 특성에 적합한 종류의 그래프가 정해져 있으며, 또한 목적에 따라 다른 그래프를 써야할 수도 있다. 가령 비율 데이터는 일반적으로 원이나 도넛 그래프를 많이 쓰지만, 시간에 따른 비율의 변화를 보여주기 위해서는 꺾은선 그래프가 더 적합하다.

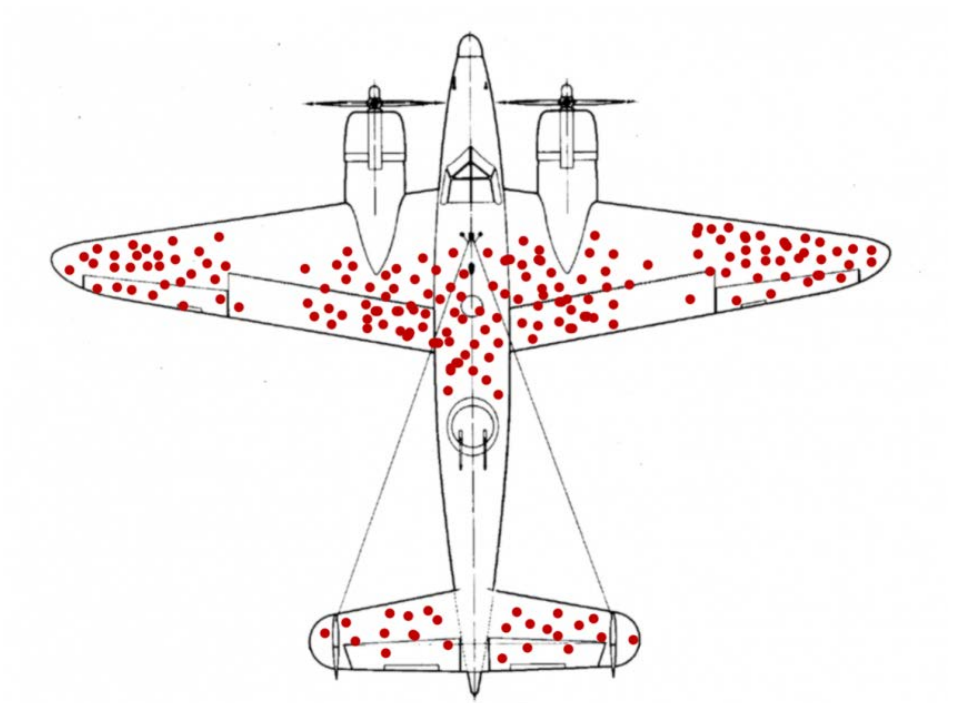
또한 축과 단위를 정하는 것도 중요하다. 그래프의 각 축에 어떤 것을 나타낼지, 그리고 그 단위를 어떻게 할지는 그래프의 명확성을 결정하는 핵심요소다. 때로는 원래 값 대신 로그를 씌우거나 백분율값을 이용하는 것도 답이 될 수 있다.

그래프는 편향된 정보를 전달할 수도 있다. 특히, 길이와 면적이 문제가 될 수 있는데 이는 길이가 2, 3배 늘어날 때 면적은 4, 9배 증가하기 때문으로 이를 고려하지 않으면 잘못된 비례의 그래프를 그리게 된다.

6. 데이터와 오류

#편향 #오류 #왜곡 #상관관계
#인과관계 #시각화오류

전투기의 피탄 흔적



2차대전 중 미 해군에서는 전투기의 생존성을 높이기 위해 작전 중 총알에 맞고 살아 돌아온 비행기들의 총알자국의 분포를 조사하였다. 위 그림의 빨간 점이 바로 그 분포로, 이에 따라 군 당국은 빨간 점이 밀집된 부분에 장갑을 더 강화해야한다고 생각하였다.

그러나 연구를 이끈 아브라함 발트란 통계학자는 이와 반대로 빨간 점이 없는 곳에 장갑을 둘러야 한다고 주장하였다.

이는 조사의 대상이 “총알에 맞고 살아 돌아온 비행기”들이므로, 저 점들은 다시 말해 맞아도 살 수 있는 위치들을 의미하며 도리어 점이 없는 곳일수록 맞으면 치명적인 부위라는 것을 의미하기 때문이다.

데이터 분석에서의 오류

데이터 분석 과정에는 항상 수많은 오류를 범할 가능성이 있다. 완벽한 분석을 하는 것은 불가능할지라도, 잘못된 결론을 내릴 수 있는 중대한 오류들에 대해 미리 파악하고, 대처하는 것은 데이터과학의 필수요소다.

데이터 자체의 오류와 편향

“인터넷 및 컴퓨터 보급 실태”에 관한 설문조사를 한다고 하자, 이 설문조사를 구글 폼으로 수행한다면 아주 편향된 결과가 나타날 것이다. 열악한 IT 환경에 있는 사람들은 해당 설문에 참여할 수 없기 때문에, 보급률은 과대평가된다.

이처럼 수집 방법, 조사 대상의 선정에 따라 데이터 자체가 이미 오류와 편향을 내재하고 있을 수 있다. 따라서 메타데이터를 분석해 이러한 잠재적 오류의 가능성을 탐색하고, 이를 제거하기 위한 방법에 대해 생각할 필요가 있다. 나아가 표본조사를 진행할 경우, 탐구하고자 하는 전체 집단에 대해 충분한 대표성을 가질 수 있는 표본을 선정해야 한다.

분석 과정에서의 잘못

데이터 자체는 문제가 없더라도, 통계 분석 기법을 잘못 적용하여 오류가 발생할 수 있다. 잘못된 식을 사용하거나, 해당 변수의 특성과 맞지 않는 기법을 사용하는 것이다. 또한 계산 및 분석 과정에서 실수로 인해 잘못된 통계량을 산출하거나, 데이터 전처리 과정에서 큰 왜곡이 발생할 가능성도 있다. 따라서 잘 알려진 샘플 데이터를 활용해 분석을 테스트하고, 제대로 된 결과가 나오는지 확인 후 실제 분석을 진행하는 방법이 유용하다.

상관관계와 인과관계의 혼동

회귀분석 및 상관계수 분석 등으로 나오는 변수들의 관계는 단순히 상관관계고 이것이 인과를 보장하지는 않으나 이들을 구분하지 못한 오류가 빈번하다.

인과의 판단

상관관계가 말해주는 것

A와 B 두 변수 사이에 양의 상관관계가 관측됐다면, 이는 A가 증가할 때 B도 증가하는 패턴이 데이터에서 나타났다는 것이며, 추가적인 조사 없이 이로부터 어떤 인과적 판단을 내릴 수는 없다.

에어컨 판매량과 선풍기 판매량에 대해 상관분석을 진행하면 분명히 강한 양의 상관관계가 있다고 나올 것이다. 그러나 에어컨 판매를 촉진시키기 위해 선풍기 할인판매를 시작해야한다고 주장하는 사람은 없을 것이다. 이들의 관계를 좀 더 정확히 파악하기 위해선 두 변수와 관련이 있는 다른 숨은 변수의 존재를 생각해야한다. 이 경우 그 변수는 기온이 될 것이다. 기온이 높아지면 에어컨의 판매가 활발해지고, 선풍기 또한 판매량이 늘어나기에 기온이 올라가고 내려가면서 에어컨과 선풍기 판매량 변수의 움직임이 같은 방향으로 나타날 것이다.

상관관계는 심증이 될 수는 있으나 확증이 될 수는 없다. 상관관계는 매우 다양한 원인에 의해 나타날 수 있다. 실제 인과관계에 의해서도, 위처럼 제3의 숨은 변수에 의해서도, 데이터 자체의 편향에 의해서도, 혹은 단순히 우연에 의해서도 나타날 수 있다. 심지어는 실제로 인과관계가 있는 두 변수가, 다른 요인에 의해 영향을 받아 통계상으로는 상관관계가 없는 것으로 나타날 수도 있다.

상관에서 인과로

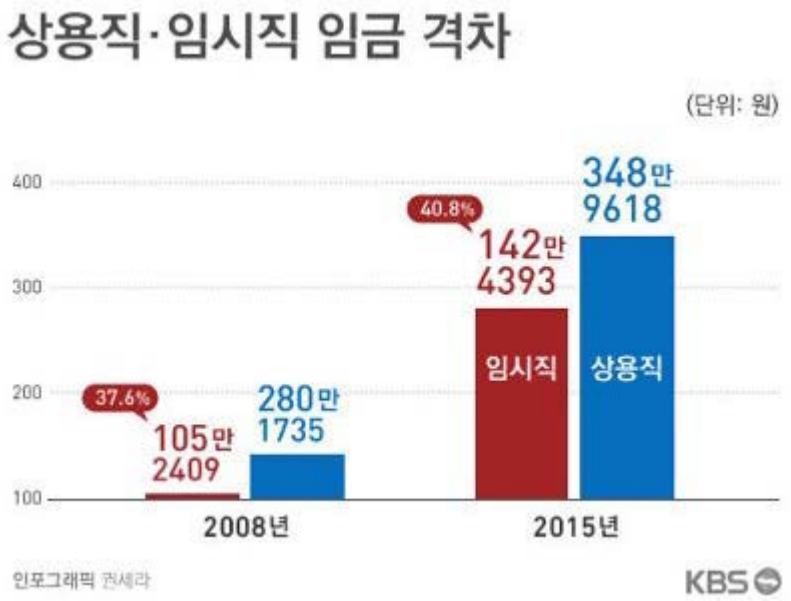
인과관계를 밝히기 위해선 나타난 상관관계에 다른 변수가 영향을 미치지 않았다는 것을 입증해야 한다. 실험실 환경에서는 실제로 다른 변인들을 통제하고 독립변수만을 변화시킬 수 있으나, 대부분의 데이터는 그런 통제가 불가능하다.

또 하나의 방법은 해당 변수를 제외한 다른 변수들을 모두 임의로 할당하거나 임의적으로 분포된 것을 밝히는 것이다. 가령 게임 플레이 시간과 점수의 관계를 알기 위해 대상 집단에게 게이밍 경험이나 성별, IQ 등의 변수와 관련없이 임의로 플레이할 시간을 할당하고 이후 점수를 분석해 나온 상관관계는 인과관계로 해석할 수 있다.

Area Principle

데이터와 분석 결과가 옳더라도 데이터를 시각화하는 과정에서 오류나 왜곡이 나타날 수 있다. 대표적인 것으로 Area Principle을 위배하는 것이다. 이 원칙은 데이터를 나타내는 도표나 차트에서 도형의 면적이 데이터의 크기와 대응해야 한다는 것으로 이를 지키지 않을 경우 오해나 혼동을 일으킬 수 있다. 그런데 역으로 보는 사람이 잘못된 해석을 내리게 하기 위해 이를 악용하기도 한다.

"매직 그래프"라는 표현이 바로 그러한 종류의 그래프를 말하는 것이다. 주로 막대 그래프에서 Y축이 0부터 시작하는 대신, 특정 값부터 시작해 전체 값에 비해서는 미미한 차이가 그래프상으로는 크게 나타나는 수법을 사용한다.



Q. 데이터 분석 과정에서의 오류, 왜곡 사례?

7. 빅데이터와 딥러닝이 만들 혁신

#빅데이터잠재력 #ERP #맞춤서비스
#연산선능 #범용인공지능

빅데이터의 잠재력

기존 데이터로 불가능한 일을 수행

사람이 쓴 글에서 감정이나 여론을 추출하는 것은 작은 데이터로는 불가능한 일이다. 같은 뜻이라도 다양한 표현이 존재하며, 비슷한 말 같아도 실제 의미는 매우 다른 경우가 많다. 이러한 경우들을 일일이 프로그램에 입력하고 설정하는 것은 불가능하나, 빅데이터를 활용하면 비슷한 것끼리 모으고 분류하는 것을 통해 감정이나 태도를 추론할 수 있다. 또한 특정 분야에서 이렇게 수집된 데이터는 다른 분야에서 수집된 데이터와 결합해 더 정확하고 자세한 분석을 가능하게 한다. 나아가 이렇게 모아진 양질의 빅데이터는 머신러닝, 딥러닝에 활용하여 더 많은 일들을 수행할 수 있다.

비용과 시간 절약

빅데이터 처리 기술을 통해 기존에는 막대한 비용과 시간이 들었을 데이터 분석을 합리적인 비용과 시간 안에 해낼 수 있게 되었다. 기업에서는 ERP와 결합해 실시간으로 의사결정에 필요한 데이터를 추출하고, 의사결정에 따른 결과를 시뮬레이션하거나 추적할 수 있게 되어 기업 규모가 커질수록 정확한 데이터를 파악하는 것이 힘들던 상황을 해결할 수 있게 되었다. 공공부문도 마찬가지로, 정책 등을 설계할 때 빅데이터를 참고하는 것을 통해 여론 수렴과 실사 검증 등의 과정을 단축하고 정확성을 높일 수 있다.

맞춤형, 개인화 서비스

개인의 생활 및 소비 패턴 등을 수집하고, 이를 빅데이터를 통해 처리한 결과와 같이 분석하는 것을 통해 개개인에게 적합한 맞춤형 서비스를 제공할 수 있다. 복잡한 설문조사를 하거나 비싼 비용을 지불하고 맞춤 서비스를 신청하지 않아도, 자동으로 수집된 데이터를 기반으로 나에게 맞는 서비스를 이용할 수 있는 것이다. 그러나 이 과정에서의 일어날 수 있는 무리한 개인정보 수집은 해결해야할 과제다.

딥러닝의 잠재력

사람이 잘 하는 것을 잘하는 기계

컴퓨터는 수치연산, 논리 판단, 알고리즘의 수행 등의 분야에서는 인간을 크게 앞서나 반대로 사람이 쉽게 하는 얼굴 및 사물 인식, 감정 파악, 직관적 판단 등에 있어서는 매우 저조한 성과를 보였었다. 우리가 무의식적으로 수행하기에 인식하지는 못하나, 그러한 행동에는 막대한 연산 능력과 빠른 정보 처리가 필요하기 때문이다.

이 때문에 인간의 뇌구조를 모사한 ANN이 등장했으며, 실제 뇌가 가진 네트워크의 규모나 처리 속도에 비해 당시 하드웨어의 낮은 스펙으로 인해 충분한 성과를 낼 수 없었으나, GPGPU와 딥러닝 기술의 발달로 실제 뇌에 비할 만큼은 아니지만 제한적인 분야에서는 인간과 비슷하거나, 뛰어난 하드웨어와 네트워크가 동원될 경우 인간보다 나은 성과를 낼 수 있게 되었다.

인간 두뇌의 성능을 초당 계산수(cps)로 나타내면 초당 1경번의 계산을 할 수 있는 수준이라고 한다. 현재 최상급의 슈퍼컴퓨터는 이 정도 성능을 넘은 3경 정도에 달하며, 쥐의 뇌는 100억 정도인데, 이것은 현재 약 100만원 정도를 가지고 살 수 있는 컴퓨터의 성능과 비슷하다. 따라서 막대한 자본을 투입한 분야에서는 이론적으로 인간보다 뛰어난 성능을 보일 수 있으며, 실생활에서도 특정한 목적을 달성하는 데에는 인간만큼 잘 할 수 있는 것이다.

범용 인공지능의 가능성

대부분의 기계학습은 특정한 데이터와 목적에만 쓰일 수 있다. 그러나 딥러닝의 경우 네트워크의 구조 자체는 다양한 데이터와 목적에 맞게 재활용될 수 있고, 실제로 딥러닝을 연구하는 기관과 기업에서는 자신의 딥러닝 솔루션을 다양한 분야에 응용하고 있다. 이러한 딥러닝의 특징은 범용 인공지능의 등장을 기대하게 만든다. 실제 인간이 다양한 분야를 학습하고 수행하는 것처럼, 잘 만들어진 딥러닝 또한 새로운 분야를 학습하고 적응할 수 있는 잠재력이 있기 때문이다.

빅데이터와 딥러닝의 활용사례

빅데이터

맞춤형 추천 및 광고 (구글, 왓챠, 페이스북 친구 추천 등)

시장 및 고객 조사 (Target의 Andrew Pole, 삼성 SSD 마케팅 등)

정책 결정(심야버스 노선 설정), 의료(약물 정보 및 유전자)

범죄 수사, 날씨 및 재난 관리, 교통(버스종합안내시스템)

머신러닝 및 딥러닝

스팸 메일 탐지 시스템, 사기/해킹 방지 시스템, 자동 교정

맞춤 광고, 컨텐츠 추천, 구글 나우

패턴 인식(문자, 필기 인식, 노래 검색, 얼굴 인식, 캡챠 등)

인공지능 작곡(Jukedeck), 애슐리 매디슨 채팅로봇, 심심이

게임 인공지능, 오토 파일럿, 기계 번역, 무인 자동차

이미지 인식(Google+ 사진 태깅, MS 아담 프로젝트, 칼로리 계산)

신약 발견, 주식 투자, 인공지능 답변조교

GAN (Generative Adversarial Network), CycleGAN

딥러닝 만드는 딥러닝

Q. **빅데이터나 딥러닝으로 인한 산업, 정책의 변화?**

나의 데이터분석 / 딥러닝 시나리오

관심분야, 주제, 문제의식

관련된 데이터

현재 의사결정 과정

데이터 수집

데이터 가공, 분석

활용

8. 정규분포와 통계적 검정

#표본 #모집단 #통계적검정 #귀무가설 #대립가설 #유의수준
#정규분포 #표준화 #Z-test #T-test

표본과 검정

검정

아침형 인간과 저녁형 인간을 비교하면 유의미한 차이가 있을까? 이러한 질문들에 대해 객관적인 답을 얻기 위해서는 통계를 사용해야 한다. 이 질문을 통계적 방법을 적용할 수 있게 수정하면 이렇게 될 것이다. “아침 7시 전에 일어나는 학생과 그렇지 않은 학생들 간에 학업성취도의 평균이 차이가 나는가?”

통계적 검정은 질문과 추측에 데이터에 기반한 답을 준다는 점에서 의사결정에 매우 중요하다. 단순히 감과 운만으로 결정을 내리기에는 현대의 문제와 선택은 너무나 복잡하다.

표본을 쓰는 이유

어떤 집단에 대한 특성을 알기 위해서 혹은 통계적 검정을 수행하기 위해 집단 전체의 데이터를 사용하는 것은 불가능하거나 매우 큰 비용이 든다. 따라서 통계에서는 연구하려는 집단에서 일정 부분만을 추출해 통계량을 계산하고, 이로부터 확률적으로 원래 집단의 통계량을 예측한다. 이들을 각각 모집단과 표본이라고 한다. 일반적으로 표본의 추출은 주관이나 편향이 개입되지 않도록 난수 등을 이용하는 Random Sampling을 이용한다. 위에서 본 상관관계와 인과관계를 구분하기 위해 임의로 할당하는 것과 같은 맥락이다.

이 때 표본의 통계량, 가령 분산을 계산하기 위해서는 원칙대로는 모집단의 평균을 알고 있어야 하나 이는 불가능하므로, 표본의 평균으로 대체한다. 나아가 분산의 경우 모집단의 분산은 N개의 변량의 편차제곱합을 N으로 나눠 구하는 반면, 표본 분산은 N 대신 N - 1로 나누어 구한다. 이는 자유도라는 개념으로 설명하는데, 표본 분산의 계산에 표본 평균이 쓰이므로, 가질 수 있는 값에 제한이 없는 모집단의 변량과 달리 표본의 경우 값이 변하면서도 같은 평균을 유지해야 하기에 적어도 하나는 평균을 맞추기 위해 고정되므로 N보다 1 작은 값을 자유도로 가지기 때문이다.

통계적 검정의 과정

가설의 설정

통계적 검정은 귀무가설(Null hypothesis)와 대립가설(Alternative hypothesis)의 2가지 가설을 세우는 것에서 시작한다. 예를 들어 동전을 10번 던진 뒤 앞면이 나온 수를 셌더니 7이 나왔다고 하자. 귀무가설은 “동전의 앞면이 나올 확률이 1/2이다”이고, 대립가설은 “동전의 앞면이 나올 확률은 1/2이 아니다”가 된다.

이 때 앞면이 나올 확률이 1/2보다 크거나 작다라는 대립가설을 세우면 단측 검정, 위처럼 어느 쪽인지는 모르겠는데 하여간 1/2과는 다르다는 가설을 세우면 양측 검정이라 하며 이에 따라 p-value를 구하는 방법이 달라진다. 단측 검정은 쉽게 귀무가설을 기각할 수 있지만 연구자의 편향에 의존한다는 문제가 있다.

유의수준 설정

공평한 동전이지만 공교롭게도 10번 다 앞면이 나올 확률도 0은 아니다. 설령 100번을 던져 보아 100번 다 앞면이 나올 확률도 0은 아니다. 통계는 표본의 특성을 이용해 모집단의 특성을 확률적으로 추정하는 것이므로 절대적인 것이 없기에, 확률이 어느 정도여야 가설을 받아들일지에 대한 결정이 필요하다. 이를 유의수준이라 한다.

유의수준은 보통 95%와 99%를 사용하며, 정말 신중한 의사결정의 경우 99%를 쓰고, 일반적으로 95%를 사용한다.

p-value

통계적 검정 기법을 적용해 p-value를 구해낸다. p-value는 귀무가설이 맞다고 했을 때 관찰된 데이터가 나올 수 있는 확률이라고 볼 수 있다. 이 때 p-value가 유의수준에 비추어 작게 나온다면 귀무가설은 기각되고 대립가설이 채택된다. 95%의 경우 p-value < 0.05, 99%의 경우 < 0.01일 때 귀무가설을 기각한다. 즉 대립가설이 해당 유의수준에서 참이라고 판단한다.

정규분포

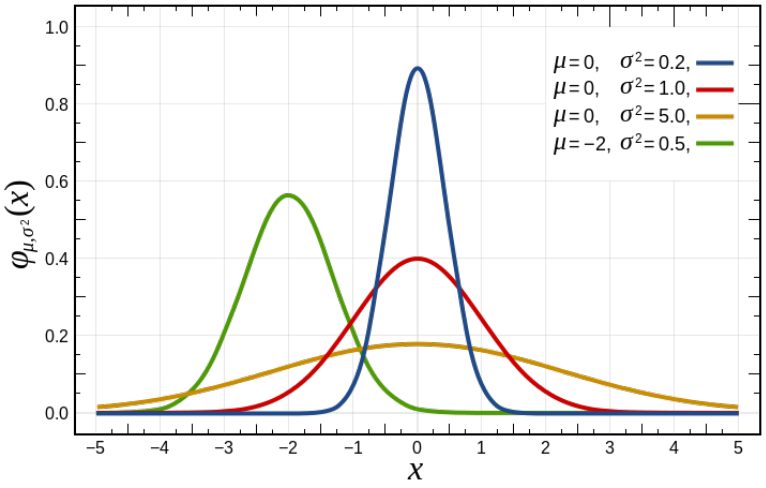
중심극한정리

평균이 μ이고 표준편차가 σ인 모집단에서 크기가 n인 표본을 반복적으로 추출할 때, 이 표본의 평균은 기댓값이 μ이고 표준편차가 $\sigma/\sqrt{n}$ 을 따르는 정규분포에 수렴한다. 또한 확률이 일정하고 독립적인 시행을 n번 반복했을 때 발생하는 사건의 평균 등 확률변수 n개의 평균은 n이 충분히 크다면 정규분포를 따른다는 것이 중심극한정리다.

말은 어렵지만 간단히 생각하면 이전의 결과가 지금의 결과에 영향을 미치지 않는 일을 여러 번 반복하면 해당하는 변수의 분포가 정규분포를 따른다는 것이다. 정규분포는 그 자체의 특이한 성질 외에도, 중심극한정리로 인해 많은 확률변수가 정규분포를 따른다는 점에서 통계학에서 매우 중요하고 자주 쓰인다.

정규분포의 모양과 성질

정규분포의 확률밀도함수는 종의 형태인데, 이 종의 중심은 곧 평균이며, 종의 경사가 완만한지 가파른지는 표준편차에 의해 결정된다.



정규분포와 표준화

평균이 0이고 표준편차가 1인 정규분포를 표준정규분포라 한다. 확률변수 x가 정규분포를 따를 경우, 다음과 같은 식을 사용해 표준정규분포를 따르도록 변형할 수 있다. 이로부터 쉽게 p-value 등을 계산할 수 있다.

$$z = \frac{x - \mu}{s}$$

z-test

위에서 구한 z 값을 이용해 바로 통계적 검정을 할 수 있다. 귀무가설을 "앞면의 확률은 1/2", 대립가설을 "앞면의 확률은 1/2이 아니다", 유의수준을 95%로 설정한 뒤 동전을 100번 던져본 것에 대해 양측 검정을 하자. 귀무가설에 따른 확률분포는 평균 50, 표준편차가 5인 정규분포가 된다. 이 때 앞면이 60번 나왔다고 하면 z값은 (60-50) / 5 = 2가 나온다. 표준정규분포표에서 이 값을 찾으면 0.977250이고 1-0.977250 = 0.02275가 바로 단측검정을 위한 p-value이다. 우리는 지금 양측검정을 하고 있으므로 2배를 해주면 0.0455가 나오며, 95% 유의수준에서 필요한 0.05보다 더 작은 값이 나오므로 귀무가설은 기각된다. 즉, 이 동전은 공평한 동전이 아닌 것이다.

t-test

n이 충분히 큰 경우, 대략적으로 n > 30인 경우 중심극한정리에 따라 정규분포에 가까워지므로 z-test를 쓸 수 있으나, 그렇지 않은 경우 t-test를 사용한다. z-test와 비슷하나 p-value를 구할 때 n의 크기에 따라 다른 값을 사용한다.

One sample t-test = 표본의 평균을 모집단의 평균과 비교하고 싶을 때

Two sample t-test = 두 집단의 평균을 비교하고 싶을 때

Paired sample t-test = 한 집단의 다른 시점의 평균을 비교하고 싶을 때

ANOVA = 비교 집단이 둘 이상인 경우 반복적인 t-test는 오류를 일으키므로 이를 방지하기 위해 ANOVA 사용

9. 회귀분석

#독립변수 #종속변수 #회귀분석 #선형회귀분석 #공선성
#OLS #결정계수 #잔차 #SSE #SSR #SST

회귀분석이란

독립변수와 종속변수의 데이터를 모아 살펴보면, 각각의 데이터마다 오차가 있긴 하나 전체적으로 봤을 때 어떤 경향성이 나타난다. 개별 데이터가 조금씩 경향을 벗어날 수는 있지만 평균적으로 이들은 그러한 경향에 돌아온다, 즉 회귀한다는 점에서 회귀분석의 어원을 알 수 있다.

장점

회귀분석은 데이터에서 변수들의 관계를 식으로 나타낼 수 있으며, 그 식이 얼마나 해당 데이터를 잘 설명하는지도 측정할 수 있다. 이를 통해 어떤 현상을 설명하거나, 새로운 데이터에 대해 종속변수의 값을 예측하는 등의 일이 가능하다. 따라서 거의 모든 데이터 활용 시나리오에서 회귀분석이 쓰이고 있다.

단점

수치적인 데이터의 값을 예측하거나 설명하는 것에는 회귀분석이 매우 절대적인 입지를 가지고 있지만, 데이터를 보고 분류하는 등 특정한 상황에 대한 확률을 측정하는 경우, 독립변수와 종속변수가 명확하게 주어지지 않는 경우, 데이터가 비정형인 경우 등에는 적용이 힘들다. 또한 실제 데이터가 회귀분석에 맞게 깔끔한 경향을 나타내는 경우도 드물며, 변수들의 관계로 오류가 발생할 수 있다.

선형회귀분석

종속변수 Y와 독립변수 X1, X2, X3, … 등이 있다고 하자. 선형회귀분석은 이들의 관계를 선형식, 즉 1차식으로 아래와 같이 표현한다.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \cdots$$

선형회귀는 해석이 쉽고, 계산이 빠르다는 점에서 장점이 있다. 선형관계가 아닌 데이터도 로그를 이용해 데이터를 변환하여 선형회귀를 적용할 수 있어 범용성 또한 높다.

선형회귀분석의 과정

변수 설정

특정한 종속변수의 분석에 사용할 독립변수들을 결정한다. 대개의 경우 독립변수가 많아질수록 예측력이 높아지지만, 모형이 너무 복잡해지거나 변수 사이의 공선성(Collinearity)로 인해 각 변수가 독립적이라는 회귀분석의 가정이 적절하지 않는 경우가 생길 수 있다. 따라서 적절한 변수를 설정하는 것이 중요하다.

또한 변수의 값을 변환시켜야 할 수도 있다. 로그를 이용해 선형관계로 만드는 것도 필요할 수 있으며, 특정한 변수가 값의 단위가 큰 경우, 전체 분석 과정에서 해당 변수의 영향력이 너무 커지기 때문에 이를 보정하기 위해 전체 변수들이 -1에서 1 등의 같은 스케일을 가지도록 조정해야 할 수도 있다. 이 때 정규분포를 활용한 표준화도 고려될 수 있다.

계수 결정

회귀분석의 오차가 최소화가 되도록 하는 계수를 찾는다. 오차를 계산하는 방법에는 다양한 것이 있으나 많은 경우 최소제곱법, 그 중에서도 OLS(Ordinary Least Squares)를 사용한다. 단순하면서 계산이 빨라 가장 많이 사용된다. 이는 측정값(실제 데이터)에서 함수값(회귀분석에 의한 예측값)을 뺀 것을 제곱하여 오차를 구하며, 이 오차의 합이 최소가 되는 계수를 찾아 구한다.

평가

회귀분석 이후 해당 데이터를 식이 잘 설명하는지 파악하기 위해 결정 계수 R^2의 값을 계산한다. 결정 계수는 0부터 1 사이의 값을 가지며 전체 데이터의 분포 중에서 회귀식이 설명할 수 있는 분포의 비율을 의미한다. 따라서 1일 경우 데이터가 완벽하게 회귀식의 그래프 위에 있는 것을, 0이면 완전히 맞지 않는 경우가 된다. 나아가 회귀식에 대해 F 검정, 계수에 대해 T 검정 등의 통계적 검정을 진행하고, 잔차(데이터와 예측값의 차이)가 정규분포를 따르는 지 확인한다.

회귀분석 공식과 용어

표기

통계학에서 변수를 보면 머리에 뭘 쓰고 있는 경우가 있다. 뜻은 다음과 같다.

Y_i=i번째 변수의 값, $\bar{Y}$=평균, $\hat{Y}$=예측값

잔차 (Residual)

종속변수의 예측치와 실제 관측치(데이터의 값)의 차이를 잔차라고 한다. 이 잔차의 제곱 합을 최소화시키는 계수를 찾는 것이 최소제곱법, 곧 OLS가 회귀식을 구하는 방법이다. Y = a + bX라는 회귀식이 있을 때 잔차는 다음과 같다.

$$r_i = \left|Y_i - \hat{Y_i}\right| = |Y_i - a - bX_i|$$

SSE, SSR, SST

위의 잔차의 제곱 합을 구한 것을 SSE (Sum of Squared Error)라고 한다.

$$\text{SSE} = \sum \left(Y_i - \hat{Y_i}\right)^2$$

이는 실제 값과 예측된 값의 차이가 얼마나 되는지를 나타낸다. 한편 예측된 값이 실제 데이터의 평균과는 얼마나 차이가 나는지를 SSR (Sum of Squares due to Regression)이라 하고 아래와 같이 계산한다.

$$\text{SSR} = \sum \left(\hat{Y_i} - \bar{Y}\right)^2$$

SSE가 회귀식이 설명하지 못 하는 부분을 의미한다면, SSR은 회귀식이 설명할 수 있는 부분을 의미한다. 이에 따라 SSE를 때로는 RSS(Residual Sum of Squares), SSR을 ESS(Explained Sum of Squares)라고 하기도 한다.

SST는 실제 데이터가 평균으로부터 떨어진 정도를 의미하며 다음과 같다.

$$\text{SST} = \sum (Y_i - \bar{Y})^2$$

결정 계수 (R squared)

OLS에서 SST = SSE + SSR가 성립한다. 이 때, 결정 계수는 다음과 같다.

$$R^2 = \frac{SSR}{SST} = \frac{SST - SSE}{SST}$$

전체 데이터의 분포 (SST) 중, 회귀식이 설명할 수 있는 부분 (SSR)이 차지하는 비율이라고 이해할 수 있다. 따라서 이 값이 1이 될 경우 데이터의 모든 값을 회귀식으로 설명할 수 있는 것이다.

그래서 계수는 어찌 구하나요

결정 계수의 값을 올리려면 SSE를 최소화해야 한다. 식을 다시 쓰면

$$\text{SSE} = \sum \left(Y_i - \hat{Y_i}\right)^2 = \sum (Y_i - a - bX_i)^2$$

SSE를 b로 편미분했을 때 0이 나올 때 SSE가 최소가 된다. 이를 이용하여 열심히 미분을 하고 정리를 하면 b는 다음과 같이 구해진다.

$$b = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sum (X_i - \bar{X})^2}$$

남은 a는 다음과 같이 계산한다.

$$a = \bar{Y} - b\bar{X}$$

b의 식을 보면 아래는 분모는 X의 분산, 분자는 X와 Y의 공분산으로 놓고 N이 약분된 것이라는 걸 알 수 있다. 그러면 b가 양수고 큰 값이 나오는 경우는, X와 Y가 공분산이 양수, 즉 양의 상관관계가 있고, X가 Y에 비해 상대적으로 중심에 몰려있을 때라고 볼 수 있다.

너무 복잡해요

대개의 경우 선형회귀는 엑셀을 포함한 모든 데이터 분석 소프트웨어에서 기본으로 제공하며, 식 또한 간단하기 때문에 일반적인 프로그래밍 언어로도 만들어 쓰기 쉽다. 중요한 것은 의미이며 직접 계산을 하는 것은 필요하지 않다.

10. 조건부 확률과 베이즈 정리

조건부확률

정의

어떤 사건 A가 일어날 확률은 다음과 같이 계산할 수 있다.

$$\mathrm{P(A)} = \frac{\text{사건 } A\text{가 일어나는 경우의 수}}{\text{일어날 수 있는 모든 경우의 수}}$$

A, B라는 사건이 있을 때, 두 사건이 동시에 일어날 확률은 P(A∩B)로 표현한다. 가령 감기에 걸리고 열이 날 확률은 P(감기∩열)이 되는 것이다.

이 때, 열이 날 때 이 열이 감기에 걸려서 나는 것인지에 대해 질문할 수 있다. 즉, 어떤 조건이 주어졌을 때, 다른 사건이 일어났을 확률을 묻는 것이다. 이는 조건부확률을 이용해 대답할 수 있다. B라는 사건이 일어난 것을 알 때, 즉 B가 조건으로 주어질 때 A라는 사건이 발생할 확률은 다음과 같이 계산한다.

$$\mathrm{P(A|B)} = \frac{P(A \cap B)}{P(B)}$$

이 식을 변형하여 아래와 같은 정리를 얻을 수 있다.

$$\mathrm{P(A \cap B) = P(A)P(B|A) = P(B)P(A|B)}$$

의미

조건부확률을 통해 관측된 결과나 현상이 어떤 원인에 의해 발생했는지를 확률로 나타낼 수 있다. 원인을 H, 결과를 D라고 할 때 이와 같은 P(H|D)를 사후확률이라고 한다. 반대로 P(H)는 사전확률이라고 한다. 이는 D라는 조건을 알기 전의 일반적인 확률이라 사전확률이라고 하며, P(H|D)는 D라는 조건을 알고 난 후 즉 D라는 사건이 발생한 이후이므로 사후확률이라고 한다. 한편, P(D|H)를 우도(Likelihood)라고 하며, 원인을 알 때 결과가 나타날 확률을 말한다. 가령 감기를 원인, 열을 결과라 하면 감기 환자가 열이 날 확률이 우도인 것이다.

베이즈 정리

정의

앞의 식을 적절히 잘 사용하여 다음과 같은 정리를 얻을 수 있다.

$$\mathrm{P(H|D)} = \frac{P(D|H)P(H)}{P(D)}$$

배가 아플 때, 왜 배가 아픈지에 대해 역학 조사를 한다고 하자. 가능한 원인은 여럿이 있겠지만 우선 점심 때 먹은 식당을 용의자로 본다고 하자. 그러면 우리가 구하고 싶은 것은 P(식당|복통)이며, 이 값이 일정 수준 이상인지를 판단해 따지러 갈지 안 갈지를 결정할 것이다. 하지만 이 확률을 직접 계산하는 것은 어렵다. 베이즈 정리를 사용하면 대신 비교적 알기 쉬운 확률을 통해 계산할 수 있다는 장점이 있다. 즉, 일반적으로 배가 아플 확률, 일반적으로 식당이 잘못일 확률, 그리고 식당이 잘못일 때 배가 아플 확률을 알면 된다. 일반적인 확률은 데이터에서 직접 계산할 수 있다. 또한 특정한 결과에 대해 역으로 원인을 찾는 것은 어렵지만, 원인을 알 때 특정한 결과가 나타날 확률은 구하기 쉽다.

따라서 베이즈 정리는 특정한 사건의 원인을 추정할 때, 직접 구하기는 어려운 확률을 간접적으로 우리가 잘 알고 있거나 구하기 쉬운 확률을 통해 계산할 수 있게 한다는데 의의가 있다.

사례: 질병 검사

T와 F를 각각 어떤 질병에 걸린, 걸리지 않은 사건이라고 하자. 그리고 D를 질병 검사가 양성인 사건이라고 하자. 그러면 질병검사를 받고 양성 판정을 받은 사람이 실제로 질병에 걸렸을 확률은 다음과 같다. P(D|Y)=0.98, P(D|N)=0.05, P(Y)=0.03, P(N)=0.97이라 주어질 때 해당 확률은 어떤 값이 나오는가?

$$\mathrm{P(Y|D)} = \frac{P(D|Y)P(Y)}{P(D)} = \frac{P(D|Y)P(Y)}{P(D|Y)P(Y) + P(D|N)P(N)}$$

나이브 베이즈 분류(Naïve Bayes Classification)

베이즈 분류는 특정한 데이터의 패턴에 대해 해당 패턴의 원인을 파악하는 기법으로, 보통 스팸 메일의 분류에 많이 사용된다. 어떤 단어들의 나열을 보고, 이 단어들이 과연 스팸 메일이기 때문에 나타난 것인지, 스팸 메일이 아닌 것인지 파악하는 것이다. 즉 P(스팸|단어들)와 P(정상|단어들)을 비교해보는 것이다.

독립과 나이브

어떤 두 사건이 독립이라는 것은, 각각의 사건 발생 유무가 다른 사건의 발생 확률에 영향을 미치지 않는다는 것을 의미한다. 가령 동전과 주사위를 던질 때 앞면과 2가 나올 확률은 서로 영향이 없기에 독립이다. 독립인 경우 두 사건이 같이 일어날 확률은 단순히 각자의 확률을 곱해서 구할 수 있다.

단어1과 단어2가 나타나는 사건이 서로 독립일 때, 두 단어가 나타났을 때 스팸 메일일 확률은 위의 성질을 이용하여 다음과 같이 나타낼 수 있다.

$$P\left(\text{스팸}\middle|\text{단어1, 단어2}\right)=P\left(\text{스팸}\middle|\text{단어1}\right)P\left(\text{스팸}\middle|\text{단어2}\right)$$

이는 단어1, 2뿐만 아니라 3, 4, 5, 등 독립이기만 한다면 계속해서 적용된다. 즉, 복잡한 결과에 대한 원인을 추정할 때도 단순히 한 결과에 대한 원인의 확률만 계산할 수 있으면 되는 것이다.

물론 이는 상당히 나이브한 가정인데, 왜냐하면 각각의 단어가 나타나는 사건이 실제로 독립인 경우는 거의 없기 때문이다. “바다”와 “이야기”라는 단어가 서로 독립이라고 보긴 힘들다. 또한 “경마”라는 단어는 그와 함께 “신규가입” 혹은 “안전” 같은 단어가 나오면 스팸일 확률이 더더욱 증가하지만, “성심”, “반대”, “서명” 등의 단어가 같이 나오면 스팸일 확률은 더더욱 감소한다.

나이브 베이즈 각 사건을 모두 독립이라고 가정하기 때문에 나이브하며, 한계가 존재한다. 그럼에도 불구하고 복잡한 문제를 간소화하고 나름대로 좋은 결과를 얻을 수 있다는 점에서 활용도가 높다. 특히, 확률 추정이 정확하지 않아도 분류 자체는 적당히 잘 하면 되기에 실용적으로는 더 높은 정확도를 보여줄 수 있다.

마르코프 연쇄(Markov Chain)

오늘의 날씨를 알 때, 내일, 모레, 3일 후의 날씨에 대해 예측한다고 하자. 미래의 날씨 예측은 당연히 이전의 날씨가 어떠했는가에 따라 결정된다. 이와 같이 어떤 시스템의 상태가 시간에 따라 확률적으로 변해가는 과정을 확률과정이라 하며, 이 중 그 확률이 오직 바로 전의 상태에만, 즉 내일의 날씨가 오늘의 날씨에만 영향을 받고 어제나 엊그제의 날씨에는 영향을 받지 않는 경우를 마르코프 과정이라고 한다.

마르코프 과정을 통해 오늘로부터 특정일 이후의 날씨에 대한 확률을 쉽게 계산할 수 있다. 가령 다음과 같은 표가 있다고 하자.

오늘 / 내일	맑음	비
맑음	0.9	0.1
비	0.4	0.6

이 때, 오늘이 맑았을 때 모레가 맑을 확률을 구해보자. 전체 경우의 수를 먼저 구하면 (1) 맑음-맑음-맑음, (2) 맑음-비-맑음이 될 것이다. (1)은 0.9*0.9*0.9 = 0.729이고, (2)는 0.081로 더하면 0.81이 나온다. 반대로 오늘 맑을 때 모레가 비가 올 확률은 1-0.81=0.19가 될 것이다.

이를 일반화하여 행렬로 나타낼 수 있다.

$$\mathbf{P^2} = \begin{bmatrix}0.9 & 0.1\\ 0.4 & 0.6\end{bmatrix}\begin{bmatrix}0.9 & 0.1\\ 0.4 & 0.6\end{bmatrix} = \begin{bmatrix}0.81 & 0.19\\ 0.6 & 0.4\end{bmatrix}$$

따라서 오늘 비가 왔을 때 모레가 맑을 확률은 0.6이라고 구할 수 있다.

미래의 상태가 과거가 아닌 오직 현재의 상태에만 영향을 받는다고 가정한다는 점에서 마르코프 연쇄도 나이브 베이즈 분류처럼 한계가 있으나, 그와 마찬가지로 복잡한 예측과 확률 계산을 간단한 과정을 통해 수행할 수 있고, 그럼에도 불구하고 적당히 괜찮은 결과를 내준다는 점에서 활용도가 높다.

11. 분류

분류란

회귀분석과 마찬가지로 분류도 독립변수로부터 종속변수를 예측하는 기법이다. 그러나 회귀분석은 종속변수가 수치적인 값을 가지는 반면, 분류는 종속변수가 범주 데이터라는 것이 다르다. 성별, 직업, 나이 등으로부터 소득을 예측할 때, 연봉을 1000만, 2000만과 같은 수치로 예측값을 낼 경우 회귀분석이지만 2000만 이상, 2000만 이하와 같이 범주로 구분할 경우 분류가 된다.

장점

회귀분석과 달리 범주 데이터를 처리할 수 있으며, 독립변수들의 값을 통해 해당 데이터가 어떤 범주에 속할지를 판단할 수 있다. 데이터 분석 시나리오의 관점에서 볼 때 실제 기업이나 사회의 의사결정에서 필요한 것은 정확한 수치가 아닌 "좋은가, 나쁜가", "A? B? C? 어디에 속하지?"와 같은 질문에 대한 답이기 때문에 많은 서비스가 분류를 활용하고 있다.

단점

분류 기법을 적용해야 할 상황은 대개의 경우 독립변수의 수가 더 많고, 다양성이 더 크다. 따라서 데이터 전처리 과정이 더 필요하고, 모형이 복잡해질 가능성이 크다. 특히 독립변수의 종류에 비해 활용할 수 있는 데이터의 양이 상대적으로 적은 경우도 있어 모형 설계 및 학습에 이를 유념할 필요가 있다.

과정

시나리오 목표 달성을 위한 종속변수를 정의한 뒤, 관련된 독립변수를 찾는다. 회귀분석과 마찬가지로 가능한 모든 변수들에서 중요한 변수를 찾아 최적화하는 작업이 필요하다. 그러나 텍스트 분류, 영화 추천 등에서의 작업에서는 독립변수의 수가 매우 늘어날 수 있다. (각 단어의 출현 여부나 영화 관람, 만족 여부 등) 이후 데이터의 수와 독립변수의 수를 고려하여 모형을 정하고, 훈련한 뒤 모델을 평가하여 가장 좋은 것을 선택한다.

분류 기법

로지스틱 회귀

일반적인 선형회귀는 Y = aX + b와 같은 식으로 종속변수 Y의 값을 예측한다. 이때 X의 값의 변화에 따라 Y값은 (−∞, ∞)의 범위를 가진다. 그런데 분류의 경우, Y는 어떤 범주에 해당하는가(Y = 1) 해당하지 않는가 (Y = 0)이 되기 때문에 [0, 1]의 범위가 된다. 선형회귀를 사용할 경우 Y가 이 범위에 들어가지 않기 때문에 예측을 제대로 할 수 없다.

따라서 이를 해결하기 위해 적절한 변환을 적용할 필요가 있다. 우선 등장하는 것이 오즈비(Odds Ratio)의 개념으로, 어떤 사건의 성공확률이 실패확률에 비해 상대적으로 얼마나 큰지를 의미하며 다음과 같이 계산한다.

$$odds\ ratio = \frac{p(y=1|x)}{1-p(y=1|x)}$$

이 오즈비는 그러면 (0, ∞)의 범위를 가진다. 그러나 아직 음수의 경우가 해결이 되지 않았는데, 여기에 로그를 취하는 로짓(logit) 변화를 수행한다.

$$logit = log(\frac{p}{1-p}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots$$

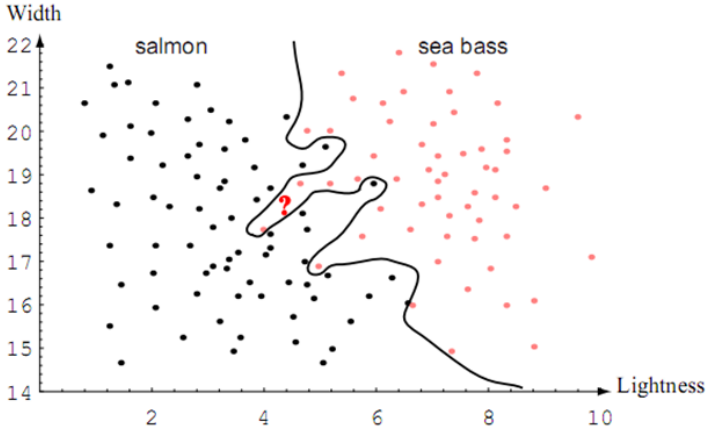
로그는 (−∞, ∞)의 범위를 가지기 때문에, 이제 좌변과 우변의 범위가 같아졌다. 이후 p, 즉 해당 x에 대해 y가 1에 해당할 확률을 구하려고 정리하면

$$\frac{e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots}}{1+e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots}}$$

이 된다. 이 함수는 0부터 1까지의 값을 가지며, 그래프 상에서는 S자 형태로 나타난다. 이러한 특징의 함수를 Sigmoid function이라고도 한다. 여기에 오차함수를 정의하고, 최적화를 통해 계수들을 구한다.

계수를 구하고 난 뒤, 실제 분류에서는 0.5와 같은 특정한 값을 기준으로, 함수에 독립변수들을 넣고 계산한 값이 그것보다 크면 Y = 1, 작으면 Y = 0으로 간주하여 분류를 진행한다. 이 기준치는 보통 0.5를 사용하지만, 상황에 따라 다른 값을 쓸 수도 있다.

오버피팅과 정규화



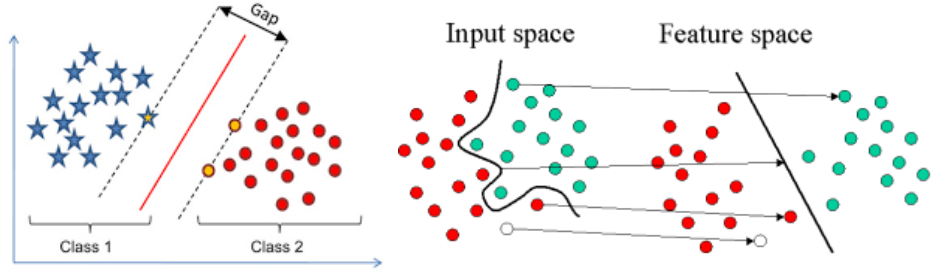
분류기의 학습을 많이 진행할수록 보통 좋은 결과가 나타나지만, 이를 실제로 적용했을 때 이전보다 더 좋지 않은 결과가 나타날 수도 있다. 이는 대부분의 머신러닝, 딥러닝 알고리즘에서 흔히 나타나는 오버피팅(over fitting)현상 때문이다. 훈련 데이터 중에서 일반적인 기준에서 벗어나는 이상치(아웃라이어)에 규칙을 맞추려다 보니 위의 그림처럼 오히려 전반적인 성능 측면에서는 더 떨어지게 되는 것이다.

오버피팅을 막기 위해서는 우선 데이터를 용도에 따라 구분할 필요가 있다. 모든 머신러닝 시나리오에서는 훈련용(train) 데이터와 최종 성능 확인을 위한 테스트(test) 데이터를 구분한다. 훈련용 데이터를 가지고 성능을 측정하면 오버피팅을 하든 무엇을 하든 훈련만 완벽하게 하면 점수가 높게 나오기 때문이다. 마치 시험문제를 미리 다 풀어보고 시험을 치는 것과 같다. 따라서 공부할 때 쓸 문제와 시험문제를 구분하는 것이다. 그런데 이렇게 할 경우 훈련 중에 모형을 조정할 때 성능을 확인하기가 힘들다. 이를 위하 검증용(validation) 데이터를 구분한다. 즉, 문제집(train) -> 모의고사(validation) -> 수능(test)와 같은 방식이 된다. 오버피팅이 발생하면 train에서는 점수가 올라가지만 validation에서는 점수가 떨어지게 되어, test 이전에 문제를 알고 모형을 수정하거나, 아니면 충분한 점수가 확보됐지만 아직 오버피팅이 나타나기 전에 훈련을 멈출 수도 있다.

오버피팅을 막는 다른 방법은 정규화(Regularization)이 있다. 이는 계수들의 값 분포를 일정하게 만들어주는 방식이다. 본래 오차는 예측된 Y값과 실제 Y값의 차이만 반영하여 정해진 함수를 통해 계산하지만, 정규화의 경우 여기에 계수들의 상대적인 분포차이를 반영하는 항을 추가한다. 따라서 설령 예측을 잘 하더라도 계수들의 값이 들쭉날쭉하면 오차가 크게 나타나게 되는 것이다.

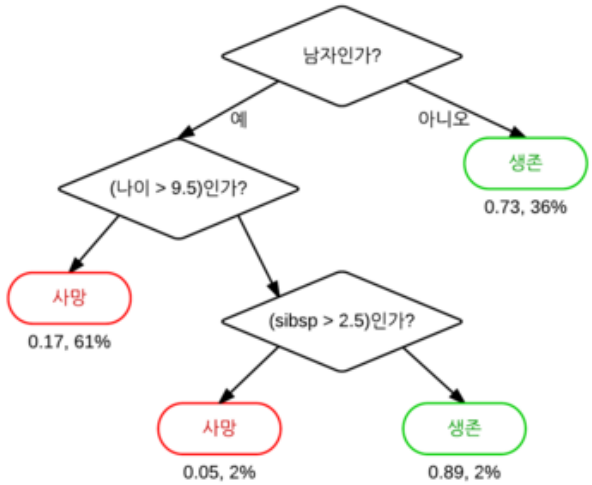
정규화에는 크게 L1과 L2가 있는데, L1은 각 계수들의 절대값의 합, L2는 각 계수들을 제곱한 것의 합이 된다. L2는 계산이 쉽고 정확한 해를 찾을 수 있다는 점에서 강점이 있다. 그러나 L1은 L2에 비해 계산은 어려우나 큰 장점이 있는데, 가령 100개의 변수가 있을 경우, L2는 변수들 사이의 계수가 골고루 퍼지게 할 수는 있어도 어떤 계수가 0이 되도록 (=그 변수를 아예 식에서 뺄 수 있게) 만들지는 않지만 L1은 가장 중요한 변수들 몇 외에 다른 변수는 0이 되도록 만드는 경향이 있어 유용하다.

SVM (Support Vector Machine)



좌측 그림처럼, 다른 부류의 데이터 사이를 가르는 어떤 경계선(벡터)를 그릴 수 있을 것이다. 이 때, 각 데이터로부터 가장 갭이 커지도록 하는 선을 찾을 수 있을 텐데 이를 Support Vector라고 하며, SVM은 이를 이용해 분류를 진행한다. 본래 SVM은 이처럼 직선으로 구분할 수 있는 경우에만 적용이 가능하나, 데이터의 좌표를 변환시키는 함수인 Kernel을 통해 우측 그림처럼 선형으로 구분이 안 되는 데이터를 이동시켜 적용할 수 있어 활용도를 높일 수 있다.

의사결정나무 (Decision Tree)



위는 타이타닉호의 탑승객 데이터로부터, 탑승객의 생존 여부를 예측하기 위한 의사결정나무의 한 예다. sibsp는 같이 탑승한 가족의 수를 의미하며, 나무의 각 끝, 즉 잎사귀 아래의 숫자는 좌측은 생존 확률, 우측은 전체 탑승객 중에서 해당 잎에 해당하는 탑승객의 비율 (혹은 어떤 탑승객이 해당 잎에 들어갈 확률)을 의미한다.

의사결정나무를 만들 때, 무엇을 기준으로 나누기 시작해야 할지를 정해야 한다. 알고리즘에 따라 세부적인 기준은 달라지지만, 공통적으로 생각해 볼 수 있는 것은 나누기 이전과 이후에 분포가 얼마나 극적으로 달라지는가를 생각해볼 수 있다. 가령 타이타닉호의 생존자 예측을 할 때, 처음 데이터 분포가 반반, 즉 생존 확률과 사망 확률이 둘 다 50%인 경우를 생각해보자. 이 때 남자를 기준으로 나누면 위처럼 50:50에서 64:36으로 갈리게 되면서 분포가 극적으로 변하지만, 이름의 길이가 홀수인가?와 같은 기준으로 나누면 50:50에서 거의 변하는 것이 없을 것이다. 따라서 남자라는 변수는 판단에 영향을 끼치는 중요한 변수지만 이름의 길이는 그렇지 않은 것이다.

의사결정나무는 다른 기법에 비해 직관적으로 이해가 쉽고, 빠르게 결과가 나오지만 정확도가 낮고 바보 같은 결과물이 나오는 경우도 있다.

랜덤 포레스트 (Random Forest)

의사결정나무는 어떤 데이터에 대해서는 매우 높은 성능을 보이다가 데이터가 조금만 달라지면 아주 낮은 성능을 보이기도 하는 등 예측력이 널뛰기하는 한계가 있다. 그러나 비슷하지만 조금씩 다른 결정 트리를 여러 개 만들어 다수결로 결과를 예측한다면 그 한계를 극복할 수 있다. (트리가 여러 개 있어서 포레스트)

물론 같은 데이터를 가지고 트리를 계속해서 만들면 같은 트리가 또 나올 것이기 때문에, 랜덤 포레스트는 부트스트랩이란 방법을 사용한다. 원본 데이터가 100개가 있다면, 이 100개의 데이터에서 중복을 허락해서 100개의 데이터를 뽑아낸 뒤, 이 데이터로 하나의 트리를 만들고, 또 다시 100개를 뽑아 트리를 만들고 반복하는 식으로 포레스트를 구성한다. 각 트리는 랜덤하게 뽑혀진 데이터에 대해 오버피팅을 하는 경우가 많지만, 트리 사이에는 어떤 편향이 없이 임의로 각자 구성됐기 때문에 전체 포레스트는 오히려 오버피팅이나 이상값 등에 대해 좋은 성능을 보일 수 있게 된다. 일종의 집단지성이라고 볼 수 있다.

랜덤 포레스트는 사용이 무척 간단하고 계산이 빠르게 진행된다는 강점이 있다. 나무를 몇 개를 만들지, 나무의 깊이는 몇까지 허용할지 등을 정하기만 하면 되는 것이다. 그러나 의사결정나무와 달리 랜덤 포레스트는 해석이 불가능해진다. 물론 포레스트 안의 나무 하나하나는 직접 보면 이해할 수 있지만, 전체 포레스트는 수 천 개의 나무로 구성됐기 때문에 이를 이해하는 것은 불가능하다.

기법이 너무 많아요!

분류 한 번 하려고 하는데 알아야 할 것이 너무 많다고 생각할 수 있다. 제일 이상적인 것은 데이터의 특징과 목표에 따라 적절한 기법을 선택하여 사용하는 것이지만, 이는 많은 경험과 이론에 대한 깊이 있는 이해를 요구한다. 그리고 그런 경험과 능력을 갖추더라도 최적의 기법을 선택하는 것은 어려운 일이다.

따라서 제일 좋은 방법은 그냥 다 해보는 것이다. 최근에는 컴퓨터의 연산 능력이 매우 발전했고, 개인도 클라우드를 통해 최고 성능의 컴퓨터를 싼 값에 이용할 수 있기 때문에, 괜찮아 보이는 기법을 다 집어넣고 한 번에 훈련시킨 다음 뛰어난 것을 골라 사용하면 된다.

12. ROC Curve

분류기 평가

회귀분석의 경우 특정한 오차함수를 지정하고, 오차의 합이나 분포 등을 통해 모형을 평가하지만, 분류는 기준치를 0.5가 아닌 다른 값을 설정하는 것 등을 통해 같은 모형이지만 다른 결과를 낼 수 있다. 이를 크게 4가지 지표의 변화를 통해 평가할 수 있다. 예측 값이 1이면 Positive, 0이면 Negative이고, 예측과 실제가 맞으면 True, 틀리면 False가 된다.

	실제 값 1	실제 값 0
예측 값 1	True Positive (TP)	False Positive (FP)
예측 값 0	False Negative (FN)	True Negative (TN)

정확도 (Accuracy)

가장 많이 쓰이지만, 데이터가 불균형(0과 1에 속하는 데이터의 양이 다름)하거나 특정한 그룹에 더 중요도가 클 경우 효과적이지 않다.

$$accuracy = \frac{TP + TN}{Positive + Negative}$$

정밀도 (Precision)

긍정적으로 평가된, 즉 어떤 사건에 해당한다고 분류한 데이터 중 실제로 그 사건이 일어난 경우를 평가한다. 가령 질병검사 결과가 양성으로 나온 사람들 중 실제로 병에 걸린 비율을 정밀도를 통해 알 수 있다.

$$precision = \frac{TP}{TP + FP}$$

재현률 (Recall)

재현률은 실제로 해당 사건이 일어난 경우 중에서, 얼마나 많은 비율을 사건이 일어났을 것이라 예측한 비율을 의미한다. 질병검사의 경우 병에 걸린 사람들 중에서 검사결과가 양성으로 나온 사람의 비율이 된다.

$$recall = \frac{TP}{TP + FN}$$

F1-Score

정확도와 재현률을 같이 고려하는 지표다.

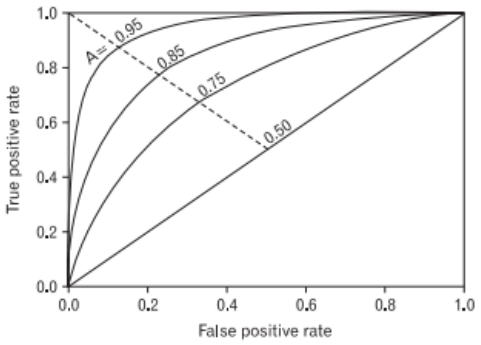
$$\mathrm{F1} = 2\frac{precision * recall}{precision + recall}$$

True Positive Rate / False Positive Rate

$$\mathrm{TPR} = \frac{TP}{postive}, \quad FPR = \frac{FP}{negative}$$

ROC Curve

ROC 곡선은 2차 세계대전에서 레이더 신호의 분석을 위해 고안됐다. ROC는 그림과 같이 가로축이 FPR, 세로축이 FPR인 그래프에서 기울기가 1인 직선보다 위에 위치한 곡선이 된다.



만약 기준치를 0.5보다 높게 잡을 경우, TPR은 올라가지만 FPR도 같이 올라가게 된다. 즉, 실제 병에 걸린 사람을 더 많이 찾지만 오진의 확률도 증가하는 것이다. 반대의 경우 오진의 확률은 줄지만 병에 걸린 사람을 찾아내지 못 하는 경우가 생길 것이다. 이처럼 TPR과 FPR은 함께 움직이게 되는데, 이 관계를 ROC 곡선을 통해 알 수 있다. 특히, ROC 곡선 아래의 면적을 AUC라고 하는데, 이 값은 0.5에서 1의 범위를 가지며, 1에 가까울수록 뛰어난 분류기라고 할 수 있다. ROC 곡선을 통해 분류기의 종합적인 성능 평과와 기준치 설정을 할 수 있다.

13. 군집분석

지도 학습과 비지도 학습

머신러닝은 학습의 방식에 따라 지도 학습(supervised learning)과 비지도 학습(unsupervised learning)으로 나눌 수 있다. 지도 학습은 입력과 그 입력에 대한 올바른 출력을 같이 제시한다. 따라서 입력을 보고 그 출력을 낼 수 있도록 학습한다. 정답을 주고 그 정답에 맞게 학습하도록 지도한다는 점에서 지도 학습이라고 한다. 지금까지 다룬 회귀분석, 분류도 입력에 대한 출력값이 주어지므로 지도 학습에 해당하며, 그 외에 많은 딥러닝, 머신러닝도 지도 학습이 많다.

그러나 지도 학습은 모든 학습 데이터에 정답을 달아주어야 한다는 문제가 있다. 가령 고양이와 강아지를 구분하는 딥러닝 네트워크를 학습시킨다고 하자. 적절한 성능을 내기 위해서는 적어도 천 개 정도가 되는 사진이 필요하다. 이 사진은 강아지나 고양이가 사진에 명확하게 나와야 하고 그리고 그 사진이 고양이 사진인지 강아지 사진인지 답이 적혀져 있어야 한다. 이 답을 데이터에 대한 라벨 (label)이라고도 한다. 그리고 이 답을 달아주는 과정을 데이터 라벨링이라고 하는데, 이는 매우 많은 시간과 비용이 드는 작업이다. 따라서 지도 학습은 모델을 설계하기 쉽고 성능도 좋게 나오지만 데이터의 질과 양에 많이 의존하게 된다는 문제가 있다.

비지도 학습은 반대로 데이터에 라벨이 필요하지 않다. 모델이 직접 데이터로부터 적절한 라벨을 판단해 붙이고 학습하는 것이다. 그러나 그만큼 학습이 느리고 부정확하다. 정확하게 라벨링된 데이터는 필요하지 않지만 데이터의 절대적 양은 확보해야 하며, 학습 과정이 전적으로 입력된 데이터에 의존하므로 데이터 자체에 결함 등이 있을 경우 잘못된 학습의 해결이 어렵거나 불가능하다.

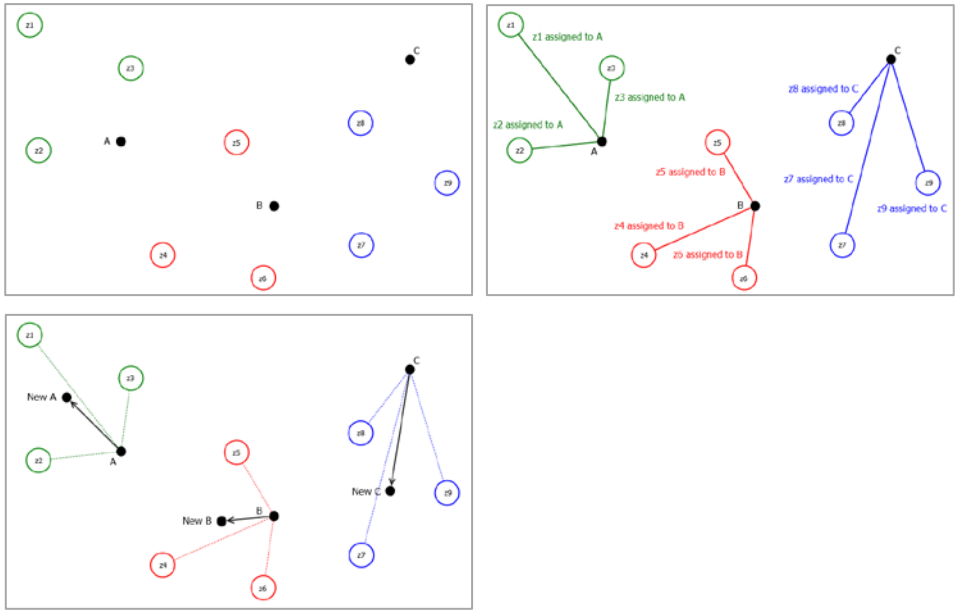
언제 지도 학습을 하고 비지도 학습을 할 지는 머신러닝의 목적과 데이터의 상태에 따라 정해야 한다. 비지도 학습에는 대표적으로 군집화(Clustering)가 있다. 입력된 데이터들을 보고 비슷한 데이터들을 그룹(군집)으로 묶는 것이며, 마케팅에서의 고객 집단 분석, 센서나 실험으로 확보한 데이터의 분석 등 다양한 곳에서 활용할 수 있다.

K-means clustering

K-means 알고리즘은 가장 많이 쓰이는 군집화 알고리즘이다. 속도가 빠르면서 방식이 직관적이고 설정이 쉽다. 특히, 각 군집의 경계를 선으로 쉽게 나눌 수 있는 경우 K-means는 가장 뛰어난 성능을 보여준다.

우선 분석에 앞서 K를 설정한다. K는 군집화에 적용할 군집의 개수이다. 만약 K=2라고 하면 2개의 군집으로 데이터를 나누며, K=5라고 하면 5개의 군집으로 데이터를 나눈다. 이 때, 적절한 K는 각 데이터의 분포에 따라 달라진다. 또한 적절한 수보다 적거나 많은 K는 성능에 큰 악영향을 미치기에, 다양한 K 값을 시도하면서 모형의 결과를 평가해야 한다.

아래의 그림을 통해 K=3인 경우의 K-means 알고리즘의 작동 과정을 살펴보자



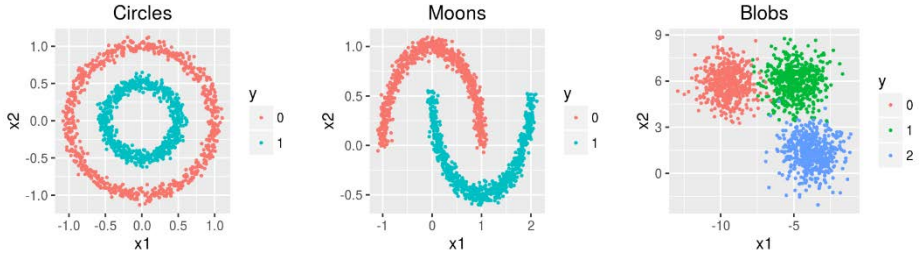
우선 3개의 점을 좌표상에 임의로 배치한다. (A, B, C가 배치됨) 이후 각 데이터들을 가장 가까운 점에 할당한다. 그리고 각 점 A, B, C들을 자기에게 할당된 데이터들의 무게중심으로 이동시킨다.

이후 이를 반복한다. 점들이 자기들에게 할당된 데이터들의 무게중심으로 이동하면서 데이터와 점 사이의 거리가 바뀌고, 이에 따라 할당 상태도 변하게 된다. 그로 인해 무게중심이 다시 변하고, 또 이동하고 할당하는 과정을 반복한다.

어느 정도 반복되고 나면 K-means는 빠르게 수렴하며, 결국 무게중심이 변하지 않는 상태에 진입한다. 이후 알고리즘은 종료되고 데이터는 각 점들에 할당된 상태가 된다. 또한 종료 시점의 A, B, C의 좌표를 통해 해당 군집을 대표하는 좌표를 알아낼 수도 있다.

K-means vs DBSCAN

K-means는 결과적으로 각 점에 대한 데이터들의 거리를 최소화하는 형태로 군집을 구분한다. 한 점에 대해 거리가 같은 점들을 이으면 원이 된다. 따라서 실제 데이터의 그룹이 좌표상에 원형으로 뭉쳐 있는 경우 가장 잘 작동한다. 그러나 그렇지 않은 경우 K-means는 잘 수렴하지 않으며 잘못된 결과를 낸다.



위와 같은 dataset이 있으면, K-means는 Blobs 유형에서만 잘 작동하며 나머지 두 유형과 같이 군집이 원형 덩어리가 아닌 경우 제대로 분석하지 못한다. 이런 경우는 K-means가 아닌 DBSCAN(Density-based spatial clustering of applications with noise)를 사용한다. 이는 서로 붙어있는, 즉 밀도가 높은 점들을 쭉쭉 이어가며 군집을 형성하는 방식이기에 같은 군집의 데이터가 이어져 있는 한 제대로 분류를 진행할 수 있다. 그러나 오히려 Blobs에서는 과하게 큰 군집을 만들어내는 등의 문제가 있다.

[활동] 모찌네 치킨집

모찌는 착실히 저금한 돈으로 치킨집을 차렸지만 장사가 처음이라 궁금한 것이 많은가 봅니다. 모찌의 궁금증을 해결하기 위해 필요한 분석 시나리오를 적어보세요. 데이터는 모찌가 착실하기 때문에 매 주문마다 주문날짜와 시간, 주문내용, 주소, 거리를 꼼꼼히 기록했다고 합니다.

주문기록을 여기저기 써놓아서 잘 파악이 안 되고 가끔 중복이 있는 모찌!

주말에는 평균적으로 주문이 더 많은 거 같은데 확신이 안 서는 모찌!

예전보다 장사는 잘 되는 것 같은데 점점 튀김기가 모자라는 모찌 언제쯤 새 튀김기가 필요할지 알 수 있는 모찌?

앞으로 단골이 될 손님인지 아닌지 알 수 있는 방법이 있는 모찌?

배달 모찌들이 나눠서 맡을 구역을 어떻게 정하는 모찌?

14. 인포그래픽

인포그래픽이란

Information + Graphic으로, 보는 사람이 이해하기 쉽도록 정보를 가공해 시각화한 것을 뜻한다. 사람들에게 노출되는 정보의 양과 복잡도는 갈수록 증가하는 반면, 이를 처리할 시간과 자원은 한정되어 있기 때문에 인포그래픽의 중요도는 더 높아지고 있다. 잘 가공되지 않아 이해가 어려운 정보는 사람들이 굳이 시간을 들여 보려고 하지 않기 때문에, 지식 전달은 물론 언론, 마케팅, 캠페인 등 다양한 분야에서 좋은 인포그래픽을 제공하는 것에 대한 투자가 늘고 있다.

좋은 인포그래픽의 요건

인포그래픽은 우선 정보를 누구나 알아보기 쉽게 표현해야 한다. 독자가 알아야 할 정보를 눈에 띄는 곳에 놓고, 크기나 비례, 순서, 인과 관계를 한 눈에 볼 수 있도록 다양한 그래픽을 사용해야 한다. 또한 여러 정보를 종합해 한 페이지에 적절한 배치로 보여줘야 한다. 또한 독자가 오해할 수 있거나 해석할 때 시간이 걸릴 수 있는 부분, 필요한 배경지식에 대해서도 고려해야 한다.

좋은 색과 디자인적 요소를 활용하는 것도 좋은 인포그래픽의 조건이긴 하나, 이에 치중한 나머지 본래의 목적을 잊어서는 안 된다. 심미적 요소는 결국 독자의 관심을 유도하고, 독자의 이해를 돕기 위해서 사용되어야 한다.

또한 매체의 특성을 고려해야 한다. 인포그래픽이 지면에 삽화로 쓰일지, 쓰인다면 어떤 크기로 들어가는지, 인터넷에 들어간다면 가로 폭은 어느 정도인지, 모바일인지 웹인지 등 실제 독자가 인포그래픽을 볼 환경에 대해서 고려하고 레이아웃을 정해야 한다.

인포그래픽을 만들 수 있는 포토샵 등의 도구를 잘 다루는 사람은 많다. 그러나 인포그래픽을 만들기 위해서는 해당 분야에 대한 지식과 데이터 시각화에 대한 지식, 나아가 가능하면 인간이 시각 자료를 이해하고 분석하는 과정에 대한 지식 즉 인지과학에 관련된 내용도 알 필요가 있다.

인포그래픽 제작 과정

목표 설정

인포그래픽을 통해 어떤 정보를 어떤 사람에게 보여줄지를 설정하고, 전달해야 하는 메시지가 무엇인지 생각한다. 또한 어떤 매체에 게시할지도 결정한다.

정보 수집 및 선택

필요한 정보를 수집한다. 데이터 및 데이터 분석 결과, 텍스트, 이미지 등 다양한 정보를 이용하되, 효과적인 전달을 위해 선택과 집중의 원칙을 활용한다. 또, 정보를 따로따로 배치하는 대신 조합하거나 연결할 수는 없는지, 쉽게 알아볼 수 있으면서도 정보량은 높일 수 있는지를 고려한다. 차트 및 시각회의 종류도 이 때 결정한다.

레이아웃

매체에 적합한 크기를 정하고, 이미지에 제목 및 내용을 배치할 방법을 정한다. 중요한 정보가 먼저, 크게 나타나야 하며, 구분이 명확해야 한다. 가로와 세로 등의 방향을 의미있게 활용해도 좋으며, 다양한 정보를 한 면에 제시할 경우는 그리드 레이아웃, 연관 있는 개념들을 같이 보여줄 경우 마인드맵처럼 중심에 주제를 놓고 주변부에 다른 정보를 배치할 수도 있다. 구조나 도식도를 이용할 경우 전체 일러스트를 중심으로 세부설명을 배치하는 것도 효과적이다.

디자인

인포그래픽에 사용할 디자인 요소를 결정한다. 어떤 색을 사용할지, 폰트는 어떤 것을 쓸지, 아이콘이나 픽토그램, 차트는 어떤 것을 쓸지, 배경에 이미지를 활용할지 등을 실제 디자인 시안을 보며 결정한다.

전문 디자이너들은 포토샵, 일러스트레이터 등을 이용해 작업하지만 파워포인트나 일반인을 위한 툴도 많이 나와있다. (망고보드 등)

15. 텍스트 마이닝과 정규표현식 (Feat. Python)

텍스트의 처리

컴퓨터가 이해하고 계산할 수 있는 수치 데이터와 달리, 자연어 데이터는 분석을 위해 더 복잡한 처리가 필요하다. 분석과 처리의 어려움에도 불구하고, 많은 기업들이 관련 연구와 서비스를 확대하고 있는 것은 오로지 텍스트에서만 얻을 수 있는 정보들의 가치 때문이다.

상품에 대한 만족도를 수치로 알아내는 것은 쉽지만 그 수치의 이유에 대해서는 알 수 없다. 이를 위해서는 상품평 데이터를 분석해 어떤 요소가 호평을 받고, 어떤 요소에 불만이 있는지 파악해야 한다. 빅데이터나 딥러닝을 이용하는 경우도 늘어나는데, 이 때에도 아래와 같은 처리가 선행되어야 한다.

토큰화

글을 이해하기 위해서는 문장을 이해해야 하며, 문장을 이해하기 위해서는 문장이 어떤 단어들로 구성되어 있는지 알아야 한다. 텍스트에서 단어를 추출하는 것을 토큰화라고 한다. "나는 사과를 먹는다."라는 문장은 "나", "사과", "먹다"라는 토큰이 있는 셈으로, 토큰화는 문장에서 문장부호나 조사, 관사 등을 적절히 제거하고 형태소를 분석해 의미를 가지는 기본 단위인 토큰을 뽑아내는 과정을 의미한다.

토큰화는 우선 텍스트에서 문장과 단어를 구분하는 것에서 시작한다. 잘 정리된 텍스트 파일의 경우 줄 단위로 문장이 구분될 수도 있지만, 대개의 경우 그렇지 않다. 따라서 문장부호 및 양식 등 다양한 정보를 활용해 문장의 시작과 끝을 판단한다. 영어의 경우 몇몇 예외를 제외하면 공백을 기준으로 단어를 구분할 수 있으며, 대소문자의 경우 모두 소문자로 체크하는 등의 방법이 있다. 그러나 이 경우에도 더 정확한 분석을 위해 단수형과 복수형 등을 구분해야 한다.

국어 등의 굴절어는 용례에 따라 어미가 매우 다양하게 변화하므로 토큰화를 진행할 때 매우 어려움이 많아 단어를 파악하는 것 자체가 힘들며, N-gram, Word embedding 등의 기법을 적용하는 것도 어렵다.

N-gram

문장 내에 어떤 단어가 있는지도 중요하지만, 단어가 어떤 순서로 놓여있는지, 단어의 앞뒤에는 또 어떤 단어가 있는지도 매우 중요하다.

N-gram은 이를 단어를 N개의 뭉치로 잘라서 저장하는 것을 통해 위와 같은 질문에 대한 답을 알 수 있게 한다. "Put a piece of kimchi on the ham"이란 문장에서 크기가 2인 N-gram, 즉 Bigram을 만들면 "(시작) Put" / "Put a" / "a piece" / "piece of" / "of kimchi" / "kimchi on" / "on the" / "the ham" / "ham (끝)"으로 나타낼 수 있다. N-gram 자체의 빈도를 파악하는 것도 유용한 정보를 제공하나, 이를 응용해 단어 출현의 조건부 확률 (빨간이란 단어 다음에 사과가 나올 확률 P(사과|빨간) 등)을 계산하거나 음성인식, 키워드 추출 등 다양하게 활용할 수 있다. OCR로 책을 인식해 얻은 텍스트 데이터에 기반한 Google Ngram Viewer가 현재 가장 방대한 데이터와 세세한 검색 기능을 갖추고 있다.

Word Embedding

각 단어를 매번 실제 텍스트의 형태로 저장하고 처리하는 것은 비효율적이므로, 대개의 경우 단어 하나에 숫자 하나를 대응시켜 숫자의 형태로 처리한다. 1 - 나 / 2 - 사과 / 3 - 먹다 와 같은 식으로 ID를 부여하는 것이다.

그러나 이 숫자는 자의적으로 부여된 것이기 때문에, 단어와 단어 사이의 관계나 개념을 표현하는 것은 불가능하다. 그렇다면 단어를 하나의 좌표에 대응하면 어떨까? 그러면 비슷한 단어는 뭉칠 것이고, 반대어나 유의어 등의 관계도 점과 점의 방향과 거리를 통해 표현할 수 있을 것이다. 이 아이디어를 구현한 것이 최근 각광받는 Word embedding 기법으로, 텍스트 데이터를 다루는 머신러닝의 학습효율을 놀라울 정도로 끌어올렸다.

Word embedding은 실제로는 비슷한 분포를 보이는 단어들은 비슷한 의미를 가진다는 가정을 이용한다. 구글의 Word2Vec은 단어의 앞과 뒤가 비슷한 맥락인 단어들을 가깝게 배치하며, 그 결과는 놀랍게도 Kings - King = Apples - Apple, King - Man + Woman = Queen이라는 관계가 성립하는 대응을 만들었다.

정규표현식(Regular Expression, 정규식)

어떤 특징을 가지고 있거나 어떤 규칙을 만족하는 텍스트를 찾을 때 정규표현식을 사용한다. 대부분의 프로그래밍 언어, 탐색기 및 검색엔진, 워드의 찾기 기능 등 텍스트를 다루는 많은 어플리케이션에서 정규식을 사용할 수 있다.

다른 사람의 컴퓨터 탐색기에 *.avi|mp4를 검색하면 놀라운 세계가 펼쳐진다. 이 패턴은 .앞에 아무 글자가 없거나 여럿이 있고 .뒤에는 avi나 mp4인 이름, 즉 확장자가 avi나 mp4인 파일을 찾는다. 정규식을 사용하면 이렇듯 복잡한 패턴을 쉽게 표현할 수 있다는 장점이 있다. 단, 프로그램에 따라 정규식 문법이 다른 경우가 있어 이에 대해서 사용에 앞서 찾아볼 필요가 있다.

기본개념과 문법 (http://regexr.com 기준)

| (엔터 위 ₩를 Shift 누르고) – 또는, 즉 찌개|찌게는 찌개나 찌게를 의미한다.

괄호 – 우선 순위 및 그룹화. Gr(a|e)y는 Gray와 Grey를 찾는다

? – 없거나 1개 있음. Colou?r는 Color와 Colour

* – 없거나 여럿. 으에*엥은 으엥과 으에엥 으에에에에에엥 등을 찾느다

+ – 1개 이상 / {n} – 정확하게 n회 반복 / {m,n} – m번 이상 n번 이하

[ ] – 사이의 문자 중 하나, -를 사용해 범위를 표현할 수 있다.

[1-9][0-9]*[개회원]는 1개, 10324회, 13213212원 등을 찾고 0개는 찾지 않는다.

^을 사용할 경우 부정의 의미가 된다. 즉 [^ab]는 a나 b가 들어있지 않은 것을 찾는다.

₩1, ₩2 혹은 $1 $2 등 – 해당 순서의 그룹을 의미한다.

(01[0-9])-([0-9]{3, 4})-([0-9]{4}) -> $1$2$3으로 바꾸면

011-123-4567은 0111234567로, 010-2222-3333은 01022223333이 된다.

사례: Python을 사용한 수능 영단어 카운터

```python
import re # 정규식 사용하기
input = open("input.txt", 'r') # 파일 열기
data = input.read() # 파일의 내용을 data란 변수에 저장

# 대소문자 앞파벳이 아닌 모든 글자를 공백으로 변환
data = re.sub('[^A-Za-z]', ' ', data)
# 2번 이상 반복되는 공백을 모두 공백 하나로 변환
data = re.sub(" +", " ", data)
# 모든 문자를 소문자로
data = data.lower()
# 공백을 기준으로 단어로 분할
data = data.split()

dict = {} # 단어 사전 만들기
for word in data: # 아까 나눈 단어들에 대해 반복
    # 사전에 있는 단어인지 확인
    if word in dict.keys():
        dict[word] = dict[word] + 1 # 있으면 횟수 + 1
    else:
        dict[word] = 1 # 없으면 추가

# 출력 파일을 열고 word와 word의 횟수를 줄 마다 출력
output = open("output.txt", 'w')
for word in dict:
    output.write("%s %d\n" % (word, dict[word]))

input.close() # 입력 파일 닫기
output.close() # 출력 파일 닫기
```

16. 딥러닝의 구조와 설계

딥러닝의 설계와 학습 과정

목표 설정

딥러닝을 통해 성취할 목표를 설정해야 한다. 목표를 달성했는 지의 여부를 판단할 기준지표도 이 단계에서 같이 설정한다.

입력과 출력 정의

딥러닝에 주어질 입력과 출력의 종류와 형태를 정의한다. 딥러닝의 전체적인 구조와 성능에 큰 영향을 미치기 때문에, 사용가능한 데이터와 목표를 함께 고려하며 신중하게 선택할 필요가 있다.

데이터 준비

사용할 데이터를 준비한다. 앞서 정의한 입력 형태에 맞게 데이터를 가공할 필요가 있다. 너무 큰 이미지는 막대한 성능을 요구하므로 리사이징을 하거나, 컬러를 흑백으로 바꾸는 등 딥러닝에 더 적합하도록 데이터의 형식을 바꾸어야 할 때도 있다. 또한 언어 데이터의 경우 단어장을 만들어 각 단어가 하나의 숫자에 대응하도록 하거나 Word2Vec 등의 기존 솔루션을 활용한다.

학습 데이터와 검증 데이터의 분리

딥러닝의 특성상, 학습이 오랜 기간 진행되면 실제 딥러닝의 성능과 상관없이 학습에 사용된 데이터에 한해서는 높은 성취도를 보일 수 있다. 따라서 학습에 사용할 데이터와 검증에 사용할 데이터를 분리해야 만 정확하고 객관적인 검증이 가능하다. 편향을 막기 위해 전체 데이터를 임의로 추출해 검증용으로 쓸 수도 있으며, 목적에 따라 학습과 검증 데이터가 아예 다른 종류가 될 수도 있다.

딥러닝 레이어 설계

입력부터 출력 레이어에 이르기까지 딥러닝의 구조를 설계한다. 사용하는 데이터의 종류와, 목적에 따라서 적합한 구조가 달라지기 때문에 이들의 특성을 잘 알고 조합할 필요가 있다. 이론적으로 더 깊고 넓은 네트워크일수록 더 복잡한 일을 수행할 수 있으나, 반대급부로 학습에 더 오랜 시간이 걸리고, 더 많은 데이터와 더 빠른 하드웨어를 요구한다. 따라서 목적을 달성할 수 있는 네트워크의 적절한 규모가 어느 정도인지를 생각하며 설계해야 한다.

학습

딥러닝 라이브러리를 통해 설계한 네트워크를 학습시킨다. 학습 방법과 학습 횟수 등의 변수에 따라 학습 시간과 효율이 크게 달라지기 때문에 이 또한 데이터와 네트워크에 맞게 조정할 필요가 있다. 가능한 경우 GPGPU를 지원하는 하드웨어와 라이브러리를 사용해야 하며, AWS 등 Cloud 서비스를 이용할 수도 있다.

재설계

학습 결과를 토대로 딥러닝 네트워크를 평가하고, 부족한 부분을 보완할 수 있도록 재설계한다. 특히 딥러닝 네트워크의 성능은 대개의 경우 실제 학습을 거치기 전엔 예측하기 어려운 경우가 많기 때문에 많은 시행착오가 필요하다. 시간과 비용을 절약하기 위해 이러한 시행착오 중에는 전체 데이터 중 일부만을 사용하고, 의미 있는 성과가 나오면 전체 데이터를 이용한 학습을 진행할 수도 있다.

검증

학습 데이터로 학습한 결과를 검증 데이터를 통해 검증한다. 목표 설정 단계에서 정의한 기준 지표를 통해 딥러닝의 성취도를 평가한다. 또한 오류나 미흡한 특성을 보이는 데이터의 특징을 파악해 딥러닝의 한계를 파악한다.

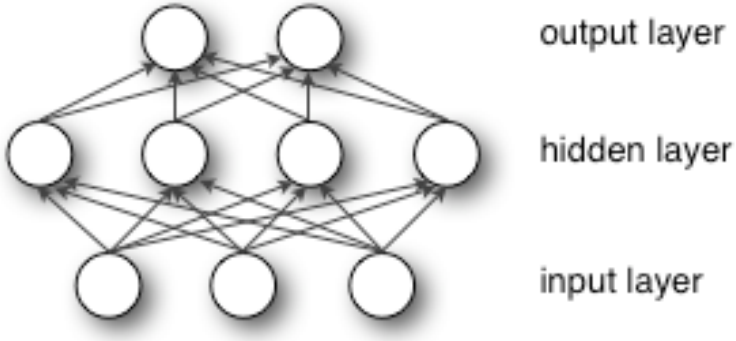
실사

학습한 딥러닝 네트워크를 통해 의미 있는 정보를 추출하거나, 별도의 서버나 하드웨어 등에 배치한다. 딥러닝은 학습은 높은 성능을 요구하지만 비해 사용에는 비교적 낮은 성능의 하드웨어로도 충분하므로, 서비스의 용도에 따라 다양한 하드웨어를 이용할 수 있다.

딥러닝 레이어

딥러닝의 구조는 딥러닝의 입력과 출력 사이에 어떤 레이어들이 쌓여 있는지를 의미한다. 각 레이어마다 다양한 특징을 가지고 있으며. 이를 잘 조합해야 데이터와 목적에 맞는 딥러닝을 설계할 수 있다.

일반적인 NN(Neural Network) 레이어 / Dense



바로 위의 네트워크의 노드(그림에서 동그라미)를 입력으로 받아 출력으로 지정된 개수만큼의 노드를 내는 레이어. 입력과 출력 노드 모두가 연결되어 있다. 이러한 NN을 많이 쌓아 DNN(Deep Neural Network)를 만들 수 있으며, 다른 레이어들과 조합하기도 용이하다.

가장 기본적인 레이어로, 분류, 변환, 회귀 등 이론적으로 충분한 수가 있으면 모든 함수를 따라할 수 있다. 그러나 계산의 효율성과 학습 측면에서의 한계도 비교적 명확하다. 따라서 다른 레이어들과 결합하여 사용되는 경우가 많다.

Activation

딥러닝 네트워크의 각 노드는 자기에게 들어오는 화살표, 즉 입력값을 각 입력에 대한 가중치를 곱한 뒤 모두 더해 계산한다. 이후 출력값을 어떻게 내보낼지를 정하는 것이 Activation 함수이다. 수치를 예측하기 위한 회귀분석이 목적이면 Linear를 쓰지만 대개의 경우 0근방에서 값이 변화가 큰 비선형적인 함수를 많이 쓰며, 최근에는 Linear에서 음수의 경우 0으로 만들어버리는 ReLU (Rectified Linear Unit)이 각광을 받고 있다.

Recurrent Neural Network (RNN)

일반적인 NN은 주어진 입력을 판단할 때 그 입력만을 고려해서 출력을 낸다. 다시 말해 맥락에 따른 판단이 불가능한 것이다. RNN은 이를 해결하기 위해 네트워크가 메모리를 가지고 이용하며, 입력한 내용이 한 번 쓰이고 끝나는 대신 다시 되새김질 되듯 네트워크에 재투입된다. 가령 1, 2, 3, 4, 3, 2, 1, 2, 3, … 으로 가는 수열을 예측할 때, 직전 숫자만 입력 받아 다음 숫자를 예측하는 것은 불가능하다. 2 다음에 3이 올 수도 있고 1이 올 수도 있기 때문이다. RNN은 2 이전의 4, 3를 같이 입력 받아 출력을 결정하기 때문에, 입력이 현재 하락하는 중이란 것을 인식할 수 있고, 정확한 예측을 할 수 있다.

따라서 이러한 맥락이 중요한 텍스트 분석이나 주가 분석, 필기체 인식 등에서 높은 성능을 보인다. 그러나 같은 크기의 다른 네트워크보다 더 많은 매개변수를 가지고 있어 학습에 시간이 오래 걸리고 난이도가 높다.

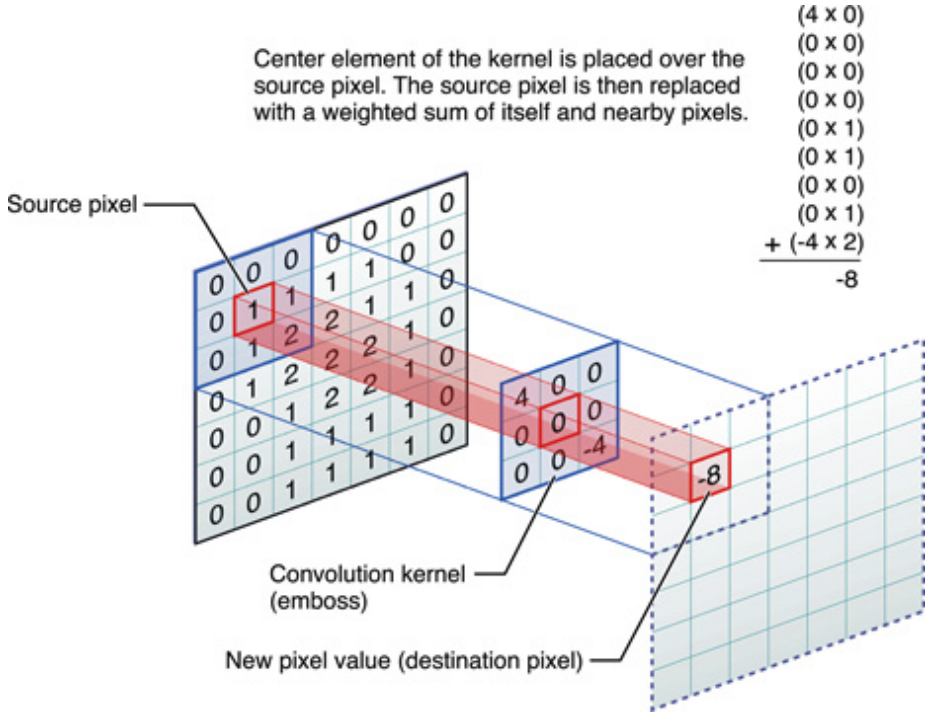
LSTM (Long Short Term Memory)

모든 딥러닝은 네트워크의 출력과 정답(목표값)을 비교하여, 정답에 가까워질 수 있는 방향으로 각 레이어의 계수를 조정하는 것을 통해 학습을 한다. 이 때 이 방향은 미분을 통해 기울기로 계산되며, 이 기울기에 일정한 학습률 (Learning Rate)를 곱하고 입력의 크기만큼 가중치를 넣어 조금씩 계수를 변화시킨다.

문제는 이 변화의 방향 혹은 가야할 방향의 경사 (Gradient)를 계산할 때, RNN의 경우 네트워크가 여러 입력에 걸쳐 출력을 계산하기 때문에 출력에서 먼 부분의 입력에 대해서는 Gradient가 점점 감소하여 0이 되거나, 혹은 점점 커지면서 무한대로 발산하는 문제가 발생한다. 쉽게 말하면 글의 내용을 인식할 때, 글 앞부분의 내용에 대해서는 학습의 방향이 제대로 전달되지 않는 것이다.

이를 극복하기 위해 뉴런 내부에 이전 뉴런의 상태를 전달할 지 말지 정하는 관문을 설치하고 이 관문의 작동도 학습시키는 LSTM이 등장하였다. 사람의 기억을 딥러닝에 구현한 것이다. 특히 LSTM은 문장의 구조적인 요소를 학습시키는데 효과가 크다. (괄호를 열고 적절하게 다시 닫아주는 등 기존의 RNN으로는 힘들었던 것을 잘 할 수 있다)

Convolution Neural Network (CNN)



컨볼루션 레이어는 특히 이미지를 인식하는 것에 큰 강점을 가지며, 사진 및 영상 데이터를 처리할 때 필수적으로 사용된다. 컨볼루션 레이어는 이미지의 각 좌표마다, 특정한 크기만큼 점과 주변 점들을 뽑아내고, 그 값에 필터를 적용한 값을 다시 내보내는 것을 통해 이미지의 다양한 특징을 추출한다.

가령 수직선을 강조하기 위해서는 다음과 같은 필터를 사용한다.

−1	0	1
−1	0	1
−1	0	1

만약 주변 픽셀이 다 같은 색이면, 한 쪽은 1, 한 쪽은 −1이게 상쇄되어 0이 나타난다. 그러나 왼쪽은 검은색(값이 0), 오른쪽은 흰색(값이 1)이라고 하자. 그러면 컨볼루션한 값은 (−1) * 0 + (−1) * 0 + (−1) * 0 + 1 * 1 + 1 * 1 + 1 * 1 = 3이며, 반대로 왼쪽이 흰색, 오른쪽이 검은색일 경우 (−1) * 1 + … + 1 * 0 + … = −3으로 수직경계선이 아닌 곳에서는 0에 가까운 값이, 수직경계선에서는 절대값이 큰 값이 나타나 구분이 가능하다.

이미지 분류에서의 사례

입력

입력받을 데이터의 크기를 정해야 한다. 가령 (128, 128, 3)의 크기라면 가로 128, 세로 128 해상도의 RGB 이미지를 입력으로 받는 것이 된다. 만약 준비된 데이터가 크기가 다른 경우, 리사이징하거나 비율이 안 맞는 부분을 흰색이나 노이즈로 채울 필요가 있다.

컨볼루션

필요한 만큼 컨볼루션 레이어를 쌓는다. 컨볼루션 레이어에서 중요한 것은 필터의 크기다. 필터가 클수록 한 점의 값을 더 많은 점을 참고해 계산할 수 있어 더 많은 정보를 참고할 수 있으나, 그만큼 더 긴 계산 시간이 필요하고, 학습에도 오랜 시간이 걸린다.

이를 해결하기 위해 Pooling 레이어를 추가할 수도 있다. Pooling 레이어는 이미지를 특정한 크기의 칸으로 나누고, 칸 안에서 Max값을 뽑아내거나 Mean값을 뽑아낸다. 만약 4 * 4 이미지에 크기가 2*2인 Pooling을 하면 총 4개 칸으로 나눠지고, 원래 4 *4 = 16에서 4개로 다루는 숫자가 줄어들면서 원래 이미지의 중요한 특징을 유지할 수 있다.

출력

컨볼루션과 풀링 이후, 최종적으로 결과를 내기 위해 NN를 마지막에 추가한다. 이 때, NN의 출력 개수는 분류의 가지수와 같아야 한다. 즉 개와 고양이, 금붕어를 구분하는 딥러닝이라면 마지막 출력의 개수가 3이어야 하는 것이다. 가령 출력이 (0.2, 0.3, 0.5)로 나왔을 경우 가장 높은 값인 0.5가 이 딥러닝의 예측이며, 이것은 금붕어라는 것을 의미한다.

학습 및 검증

전체 이미지 셋에서 학습용 데이터와 검증용 데이터를 분리한 뒤, 충분한 시간을 들여 학습한 뒤 정확도를 평가한다.

강화학습 (Reinforcement Learning, RL)

실제 세계에서 우리가 학습을 하는 방법은 아주 정확한 지도 학습도, 비지도 학습도 아니다. 답을 아는 사람도 없고 정확한 채점을 받지도 않지만, 그렇다고 답에 대한 어떤 단서조차 없는 것은 아니기 때문이다.

어떤 상황에서 특정한 행동을 하면 그에 대한 결과가 발생하고, 그 결과를 토대로 이 결과가 좋은 것인지 나쁜 것인지를 판단한다. 좋은 결과가 나왔으면 앞으로 비슷한 상황에서 그 행동을 다시 하며, 나쁜 결과라면 다른 행동을 취한다. 이렇게 경험이 축적되면서 점점 더 나은 행동을 할 수 있게 학습이 되고, 자전거를 타는 것이나 게임, 요리, 공부 등등 인간의 많은 활동이 이러한 구조를 통해 학습이 이뤄진다.

이를 기계학습에 구현한 것을 강화학습이라고 한다. 자신을 둘러싼 환경의 상태를 보고, 그 상태에 가장 적절한 행동을 취하며, 적절한 행동은 가장 많은 포상을 받은 행동이 된다. 포상은 단순히 해당 행동이 바로 발생시킬 포상만이 아니라, 그 행동 이후의 발생할 미래의 상황에서 나올 포상을 같이 고려한다. 이 때 미래의 포상을 현재의 포상에 비해 어느정도 비율로 평가할지를 정하는데 이를 할인율이라고 한다. 할인율이 0에 가까울수록 근시안적인 행동을, 1에 가까울수록 미래지향적인 행동을 한다.

문제는 이를 실제로 구현하려면, 가능한 모든 상황에 대해 어떤 행동을 취할지를 계산해야 한다는 것이다. 세로를 상황, 가로를 행동, 칸의 값은 해당하는 상황에 대한 행동의 보상을 채운 매우 거대한 표가 필요하다. 이 표를 채우려면 Q(입력, 행동) = 포상이 되는 Q 함수를 계산해야 한다. 그리고 강화학습은 이 Q 함수를 나름대로 찾아가는 과정이라 할 수 있다.

여기서 Q 함수를 딥러닝 네트워크로 근사하여 사용하는 것을 DQN (Deep Q-Learning)이라고 한다. 알파고를 만든 딥마인드가 DQN으로 고전게임을 정복하였고 결국 이를 기반으로 인공지능이 바둑을 정복하였다. 무인 자동차도 사물을 CNN으로 인식하지만 실제 운전은 강화학습을 통해 학습하며, 많은 노동과 활동이 강화학습을 통해 인공지능이 할 수 있는 일이 되어가고 있다.

GAN (Generative Adversarial Network)

사이 좋은 형제가 있다. 형의 이름은 Discriminator, 위조지폐와 진짜지폐를 구분하는 일을 한다. 동생의 이름은 Generator, 위조지폐를 만드는 일을 한다. 형제는 따로따로 살면서 각자의 생업에 종사하지만, 주말마다 모여 서로의 기술을 뽐내어 겨룬다 (Adversarial). 동생은 정말 진짜 같은 위조지폐를 만들어 진짜와 섞어 형에게 보여주고, 형은 그 사이에서 위조지폐를 가려낸다. 시간이 지날수록 형제의 실력은 일취월장한다. 형은 최고의 감별사로, 동생은 최고의 위조범으로.

GAN은 바로 이러한 체계를 딥러닝 네트워크로 구현한다. 일반적인 분류 문제를 위한 딥러닝은 Discriminator 하나를 두고, 라벨링된 데이터를 기반으로 학습을 진행하는데 GAN은 추가적으로 Generator라는 네트워크를 두고 이 둘을 동시에 학습시키는 것이다. Discriminator의 목적은 기존의 분류 모델과 같이 진짜와 가짜를 최대한 정확하게 구분하는 것이다. 반면 Generator는 Discriminator가 잘못된 답을 낼 때 점수를 얻으며, 이를 극대화하는 방향으로 학습한다.

Ian Goodfellow가 2014년 처음 제시한 이후 재조명받은 GAN은 근래에 가장 핫한 주제다. 많은 연구진과 기업들이 GAN을 이용한 프로젝트를 추진하고 큰 성과를 내고 있으며, GAN의 단점을 해결하기 위한 새로운 수학적 해법이 도입되고 더 많은 데이터가 쌓이는 등 점점 발전속도가 빨라지고 있다.

어떤 이미지를 올리면 그 이미지를 고흐나 다른 작가의 스타일로 바꿔주는 앱을 본 적이 있을 텐데, 이러한 Style transfer는 대표적인 GAN의 활용 사례다. GAN은 특히 이미지의 생성에 매우 강점을 보인다. 기존의 생성 모형은 확률을 기반으로 하기에 상당히 뿌연 이미지를 생성했는데, GAN은 매우 선명하면서 그럴듯한 이미지를 만들어낸다. 또한 생성적이기 때문에 단순히 기존 데이터를 분류하는 것을 넘어 인공지능이 새로운 데이터를 만들어낼 수 있다는 점에서 더욱 활용도가 높다.

사실와 구분하기 힘든 조작된 이미지와 비디오를 인공지능이 쉽게 만들 수 있는 미래를 생각해보자. 생각이 끝났다면 이제 그 미래가 현실이라는 것을 받아들이자. 나의 세계관은 얼마나 무너지고 변화해야 하는가?

17. 데이터과학 한걸음

데이터과학자의 자질

데이터과학자에 대한 높은 수요에도 불구하고, 그 공급은 제한적이다. 프로그래밍이나 통계, 수학을 알고 있는 것을 넘어 특별한 자질을 요구하기 때문이다.

호기심

우선 데이터과학자는 항상 호기심을 가지고 데이터를 바라볼 필요가 있다. 서로 관련이 없어 보이는 데이터들에서 놀라운 관계를 찾아낼 수도 있으며, 가치가 없다고 무시했던 데이터에서 막대한 가치를 창출할 수도 있다. 또한 현재의 상황에 만족하는 대신 더 나은 가능성에 대해 계속해서 탐구하고 시도해야 한다.

질문

이는 곧 질문을 잘 하는 능력으로 귀결된다. 데이터과학은 결국 질문을 하고 그 질문에 대한 답을 데이터에서 찾는 것이다. 잘못된 질문을 하면 잘못된 결과를 얻기에, 답을 찾는 과정보다 중요한 것이 질문이다. 또한 데이터과학자는 혼자 일하는 법이 없기에, 다른 사람의 질문을 듣고 활용하는 법도 배워야 한다. 가령 “우리 점심 뭐 먹을까?”라는 질문을 “현재 위치에서 10분 이내에 도달 가능한 범위의 음식점의 목록 중 상대가 선호하는 군에 속하는 메뉴를 제공하는 것을 골라 식사 후에 기분이 좋아질 확률이 높은 순으로 3개를 조사해 제시하라”는 질문으로 해석할 줄도 알아야 진정한 데이터과학자가 되는 것이다. 당연한 말이지만 이는 단순히 코딩, 통계, 수학만으로 가능한 것은 아니다.

협업

나아가 협업도 중요한 요소다. 데이터과학을 조직에 도입하고 문제를 해결하는 것은 혼자서 가능한 일이 아니다. 조직 전체를 설득해야 한다는 어려움을 극복해야 하며 비협조적인 이들에게 당신의 가치를 증명해야 한다. 데이터를 얻기 위해 복잡한 이해관계를 조율해야 할 수도 있고, 당신이 찾아낸 발견과 메시지를 남들도 알 수 있도록 전달해야 한다.

우리의 여정

본격적인 데이터과학자가 되기는 한참 멀었지만, 그럼에도 불구하고 많은 것을 배웠고 제한적이지만 데이터 분석을 스스로 수행할 수 있는 기반을 갖추었다. 데이터 분석 시나리오 속에서 우리가 배운 것을 어떻게 활용할 수 있는지 보자.

목표 설정

잘 알고 있는 분야에 관련된 문제를 탐색한다. 배경지식이 있으면 어떤 데이터를 어디서 찾아야할지 쉽게 알 수 있으며, 구한 데이터의 특성도 빠르게 파악할 수 있다. 또한 분석이 잘 되었는지, 문제는 없는지 파악이 가능하며, 나아가 그 결과를 어떻게 활용해야 할지도 생각하기 쉽다.

문제를 정할 때 너무 방대하게 생각하는 대신 사용할 수 있는 기법에서 출발해 해당 기법을 관심 분야에 어떻게 적용할 수 있는지 거꾸로 생각하는 것도 좋다.

수집

비정형 데이터를 다루는 것은 배운 범위 밖에 있기에, 정형 데이터 중 가능하면 표의 형태로 되어 있는 것을 찾는다. 국내 자료의 경우 네이버 데이터랩을 이용하면 편리하다. 때로는 데이터를 제공하는 사이트에서 엑셀 파일 혹은 csv 파일을 다운로드할 수 있다. 그렇지 않은 경우 내용을 복사해 엑셀에 붙여넣고 가공하거나, 30개 이하의 작은 자료의 경우 직접 표 형태로 정리하는 것도 고려할 수 있다. 필요에 따라 구글 트렌드나 Data repository를 이용할 수도 있다.

한편, 기법을 적용할 때 테스트를 하고 싶은 경우 해당 기법명 + sample data로 구글에 검색하면 테스트 데이터를 얻을 수 있다.

저장

데이터를 저장하는 것 자체는 큰 문제가 없을 것이다. 그러나 분석에 앞서 항상 원본 데이터 파일을 따로 저장하고, 분석에 사용하는 파일을 따로 두어야 한다.

가공

수집한 데이터를 분석이 가능한 형태로 가공한다. 우리나라 웹에서 얻은 자료는 대개 표 양식이 복잡한 경우가 많으므로, 안타깝지만 가능한 간단하게 분해하고 행/열 전환 등을 통해 변환한다. 데이터 출처가 여럿인 경우, 우선 각각을 공통된 형태로 변환한 다음, 새로운 표를 만들어 모든 레코드를 붙여넣고 중복값을 제거한다. 이 때 누락된 값이 있으면 일정한 규칙에 따라 처리한다. 난감한 처리도 엑셀 함수 및 필터를 사용하고, 여러 버전의 표를 만들어서 반복적으로 가공을 하면 해결이 가능하다.

수치 데이터의 경우 이 과정에서 바로 기술 통계학적 분석을 진행하면 데이터의 특성을 빠르게 파악할 수 있다. 또한 분포도나 막대 그래프 차트를 그려서 분포가 어떻게 되어있는지를 보아도 유용하다. 또한 원래 숫자를 그대로 이용하는 대신, 비율이나 변화율, 변화량을 계산하거나 로그, 지수, 정규화 등을 통해 가공하는 것도 고려할 필요가 있다.

분석

분포의 일반적 특성을 알고 싶을 때 – 기술 통계학

여러 변수 사이의 상관관계를 확인할 때 – 상관 분석

변수들의 관계를 수식으로 나타내고 싶을 때 – 회귀 분석

특정한 입력을 바탕으로 어떤 값을 예측할 때 – 회귀 분석

입력 변수를 통해 어떤 그룹이나 범주로 분류하고 싶을 때
– 나이브 베이즈 분류기, 로지스틱 회귀, SVM, 랜덤 포레스트 등

데이터의 값이나 평균을 다른 데이터와 비교할 때: z-test, t-test

확률을 기반해 미래의 상태를 예측할 때: 마르코프 연쇄

복잡하고 추상적인 작업이 필요할 때: 딥러닝

위와 같이 여러 경우에 따라 적절한 분석 기법을 선택하고 수행한다. 이후 분석 결과와 가정을 비교하고 차이가 있다면 어떤 이유가 있는지, 예상대로라면 혹시 나 잘못된 과정은 없는지 찾아보고 필요에 따라 추가적인 분석과 가공을 진행하여 보완한다. 특히, 분석 과정에서 모형을 평가하기 위한 test 데이터와 훈련을 위한 train 데이터를 꼭 분리하고, 훈련을 보조하기 위한 validation 데이터를 활용하도록 한다.

평가

분석 결과를 해석하고 평가한다. 유의수준, 결정계수, AUC 등의 대표적인 지표를 통해 모형이 데이터를 얼마나 잘 설명하는지 파악할 수 있다. 평가 결과를 토대로 새로운 모형을 만들고 비교하는 것을 통해 모형을 개선시킨다. 이후 선정된 모형을 더 심층적인 지표와 잘 선정된 test 데이터로 특성을 파악한다.

해석

모형을 통해 분석된 결과와 데이터가 의사결정에 어떤 영향을 줄 수 있는지 해석한다. 이는 초기에 설정한 목표와 조직의 현재 의사결정 과정, 상태 등에 영향을 받는다. 전체 분석의 메시지를 추출하고, 이를 뒷받침할 근거들을 정리한다.

적용

대부분의 사람들은 데이터 분석의 전문가가 아니다. 이들을 설득하고 변화시키기 위해서는 이해하기 쉽고 호소력이 있는 표현으로 메시지를 전달해야 한다.

"95% 유의수준에서 A그룹으로 분류된 소비자 집단의 판매량 평균이 B그룹으로 분류된 소비자 집단의 판매량 평균보다 높게 나왔다. A그룹과 B그룹을 구분하는 로지스틱 회귀 함수는 Y = a*변수1 + b*변수2 + … 이며 또한 OrderTerm 변수는 이들 그룹이 분산이…"

"남성 소비자 중 매 주문마다 5개 이상 구매하는 그룹이 그렇지 않은 그룹들에 비해 높은 판매량을 보였습니다. 이들은 주문 사이의 공백기간이 길고 제품의 세부사항에는 큰 관심을 두지 않습니다. 따라서 신제품에 대한 프로모션과 대량 구매에 대한 할인 혜택 등을 알리는 메일을 주기적으로 전송하면 판매량을 높일 수 있을 것입니다."

데이터 과학자가 하는 일은 전자가 아닌 후자처럼 말하는 것이다.