IDG
STRATEGIC
MARKETING SERVICES

IT WORLD CIO

IDG Deep Dive

"비즈니스 결과물로 직결되는"
AI, 머신러닝, 딥 러닝 이해와 활용 가이드

AI, 머신러닝, 딥 러닝은 2019년 현재 가장 뜨거운 관심을 모으는 분야인 동시에, 개별 기술 간 경계를 넘어 전 산업 분야에 커다란 파급 효과를 미칠 변화로 여겨진다. 그러나 실제 기업 내 AI 개발 및 활용 프로젝트가 수익 등 구체적인 성과로 연결되는 경우는 흔치 않다. 지금 가장 필요한 것은 복잡하게 연결된 각 기술의 개념 정의를 내리고 기업에서 바로 쓸 수 있는 머신러닝 알고리즘 사례를 선별하는 작업일 것이다. 여기에 더해 세 기술의 접점을 통해 기술 프레임워크와 기업의 성과를 연결하는 법, 기업 내 AI 프로젝트의 실패 원인까지 파악하면, 성과와 연결되는 구체적인 AI 활용 전략을 세울 수 있을 것이다.

- 왜 AI인가 : 비즈니스 맥락에서의 AI
- AI 정의 : BI에서 AI로의 자연스러운 진전
- AI, 머신러닝, 딥 러닝은 어떻게 다른가
- 머신러닝 모델 성능 측정 방법
- 머신러닝 모델 구축이 어려울 수 있는 이유
- AI 프로젝트가 실패하는 이유

"비즈니스 결과물로 직결되는" AI, 머신러닝, 딥 러닝 이해와 활용 가이드

Tech Trend

Jerry Hartanto | Infoworld

인공지능, 머신러닝, 그리고 딥 러닝은 2019년 현재 가장 뜨거운 카테고리이며 기술적 파생도 무수히 많다. 그러나 이들 기술에 대한 뉴스 상당수가 노이즈 마케팅이며, 대부분은 너무 모호하거나 일반적이고, 수학이나 특정 분야에 치중되어 있다는 지적도 있다. 특히 사업 성과나 비즈니스 지표(Metrics)와 단절되어 있고 목적성이 없다는 비판도 받고 있다.

본 기사에서는 AI, 머신러닝, 딥 러닝의 정의를 내리고, 각 구성요소가 비즈니스 프레임워크를 보완하는 방법, 사업 성과와 비즈니스 지표를 활성화하는 방법을 설명한다. 동시에 머신러닝과 딥 러닝 모델 학습(Model Training), 알고리즘(Algorithm), 아키텍처(Architecture), 성과 평가(Performance Assesments)의 일반적인 유형과 좋은 성과를 방해하는 장애물을 설명하는 것을 목적으로 한다.

AI를 구성하는 각 요소를 상세히 들여다보자

왜 AI인가 : 비즈니스 맥락에서의 AI

모든 조직은 구체적인 성과를 목표로 한다. 각각의 조직마다 목표를 달성하기 위해 매출, 비용, 시장 출하 시기, 프로세스 정확성, 그리고 효율성 등 여러 비즈니스 지표와 프로세스를 최대한 효율적으로 조율한다. 하지만 자금, 시간, 인력 같은 조직의 자원은 유한할 수 밖에 없다. 그래서 문제는 자원 할당이 올바르게 이루어지고 있는지, 어떤 종류의 자원을 어느 정도의 수/양으로 배치할지, 어떤 조치를 취할지, 어떤 역량이 필요한지를 고민하고, 이런 훌륭한 결정을 경쟁사보다, 또 시장의 변화 속도보다 빨리 내리는 것으로 귀결된다.

어려운 일이지만 데이터, 정보, 지식을 활용할 수 있을 때는 의사결정이 분명 훨씬 더 쉬워진다. 이런 요소를 사용할 수 있다고 가정할 때, 결실을 얻으려면 데이터, 지식을 취합해 새로운 정보를 발굴해야 한다. 애널리스트가 자신의 머리 속에만 있는 지식(Tribal Knowledge)을 끄집어내고, 변동하는 비즈니스 규칙에 맞게 조절하며, 가능하면 개인적 편견을 보정하고 패턴을 찾아내어 통찰력을 제공하기 위해서는 시간이 필요하다. 애널리스트와 관리자가(충분한 시간을 가지고) 여러 가지 시나리오를 평가하고, 권고사항과 의사결정 사항에 대한 확신을 높이기 위해 몇 차례의 실험을 할 수 있게 된다면

제리 하란토는 데이터 인텔리전스, 클라우드 솔루션, 사이버 분석, 데브옵스(DevOps)와 데이터센터 솔루션 등 다양한 분야의 공급 업체 트레이스3(Trace3)에서 AI와 셀프서비스(Self-Service) BI 프랙티스 부서를 이끌고 있다. 하란토는 경영 컨설팅, 기업/사업 전략, 마케팅과 세일즈, 운영 및 프로세스 개선, 그리고 제품 개발과 엔지니어링 관련 경험을 쌓았으며, 맥길 대학교에서 전자 공학 학사를, 존스 홉킨스 대학교에서 전자공학 석사를 받았으며, 미시건 대학교에서 MBA를 받았다. 본 기사와 수치는 2018년 11월에 개최된 사우스랜드 테크놀로지 컨버런스(SoTec)에서 가져온 것이며 트레이스3의 허가를 받고 사용되었다.

데이터 가치 사슬의 핵심, 데이터나 분석이 아니다

더할 나위 없을 것이다. 마지막으로, 이런 의사결정 사항은 현장에서 운용할 수 있어야 한다.

AI, 머신러닝, 딥 러닝을 도입할 때 다음 사항을 고려해야 한다.

- 관측 결과(Observations)를 기초로 조직을 모델링하는가
- 수많은 요인과 변수를 동시에 검토해서 얻은 통찰력 창출(한 사람이 합당한 시간과 비용 제약 하에서 얻을 수 있는 것을 초월)하는가
- 새로운 관측 결과가 제공될 때마다 계속해서 학습하는가
- 결과물의 가능성을 계량화하는가(즉, 어떤 일이 벌어질지를 예측)
- 비즈니스 목표와 지표를 최적화 하기 위한 특정 조치사항(Actions)을 사전에 정의하는가
- 기존의 더 느린 재 프로그래밍 보다 더 빠른 재 교육을 통해 새로운 비즈니스 규칙에 신속하게 대응하는가

더 낮은 컴퓨트 및 스토리지 하드웨어 비용, 그리고 도구와 결합된 데이터 볼륨과 데이터 유형의 확산은 AI, 머신러닝, 그리고 이제 딥 러닝까지 이끌어냈다. 페이스북, 구글, 아마존, 그리고 넷플릭스 등 웹 스케일(Web-scale) 기업이 이 개념이 통한다는 것을 입증했고, 이제 모든 분야의 조직이 이들 기업을 따르고 있다. 인공지능, 머신러닝, 그리고 딥 러닝이라는 3인조는 비즈니스 인텔리전스(Business Intelligence, BI)와 결합해 의사결정 과정(Decisioning)의 장애물을 극복하고 이제 <표 1>에서처럼 조직의 사업 목표 달성을 촉진하는 요소로 자리 잡았다.

AI, 머신러닝, 딥 러닝은 지표 주도형 조직 및 산업에 속한 모든 개인에도 적용된다.

맥킨지 클로벌 인스티튜트는 2011년 5월에 발간된 "빅 데이터: 혁신, 경쟁, 그리고 생산성의 다음 제약"에서 관리자 수에 비해, 분석 결과의 사용법을 알고 있는 애널리스트 공백이 150만

표 1 | AI로 비즈니스 의사결정 과정 개선하기

인공지능, 머신러닝 그리고 딥 러닝	비즈니스 인텔리전스	목표와 의사결정 과정에 대한 장애물	조직 목표
• 비즈니스 모델링: 입력을 기반으로 목표 산출물을 복제 • 사람이 할 수 있는 패턴과 통찰력 발견 • 자율 의사결정: 예측, 맞춤처방, RPA(Robotic Process Automation: 로봇 프로세스 자동화) • 재 프로그래밍 보다는 재 교육을 통해서 변화하는 비즈니스 규칙에 대응 • 중앙집중형 데이터 리포지토리(Repository), 전용 컴퓨트, 개방형 머신러닝 라이브러리 활용 • 통찰력 + 예측 + 재 교육 = 더 낮고 더 빠름	• 비즈니스 이력: 주도요인에 대한 계량화된 비주얼 지표, 벤치마크, 정밀분석 • 정밀분석으로 근본원인 분석 규명 • 의사결정 지원: 트렌딩(Trending), What-If 분석, 민감도 분석 • 애플리케이션에 비즈니스 로직을 재 프로그래밍하여 의사결정 구현 • 엔터프라이즈 전체의 데이터를 결합하고 이는 중앙집중형 데이터 리포지토리 활용	• 로컬 대 E2E 관점, 문서화 되지 않은 지식, 데이터 전처리(Data wrangling: 랭글링), 그리고 장기 분석(Slow Analysis) • 빈번하지 않은 실험 • 너무 적거나 너무 많은 데이터 • 사람의 편견(직관과 육감) • 변화하는 비즈니스 규칙은 의사결정과 프로세스 실행에 대해 비싸고 시간이 걸리는 재 프로그래밍을 필요로 한다	• 지표: 매출, 비용, 효율성, 다음과 같은 이해당사자: – 신규고객 유치, 기존고객 유지, 점유율 – 제품 개발 TTM 및 특징 – 프로세스 정확성, 일관성, 효율성, 자동화 • 자원에 대한 의사결정, 예상 산출물, 결과, 그리고 위험 • 데이터를 활용해서 더 낮고 더 빠른 의사결정

명에 이르고 있다고 기술했다. 이는 데이터 애널리스트와 데이터 과학자 등 분석 업무 수행 인력의 몇 배나 되는 수치다.

즉, 데이터 가치 사슬의 핵심은 데이터나 분석이 아니라는 의미다. 핵심은 맥락에 맞게 그리고 정교한 대응을 위한 지능적인 방식으로 데이터/분석을 소비할 수 있는 능력이다. 이것은 비

표 2 | AI가 비즈니스 프레임워크와 문제점을 보완

	프레임워크/문제	대표적인 비즈니스 질문
대고객	4P	고객은 어떤 제품 특징에 반응을 보일까? 가격은 얼마로 해야 할까, 그리고 몇 개나 팔릴까? 다양한 판촉활동에 누가 반응할까? 어떤 채널에서 최상의 결과가 나올까?
	고객 세분화	어떤 속성을 기준으로 고객군을 세분화할 것인가?
	기존고객 유지와 이탈	어떤 고객이 이탈할 가능성이 있고, 무엇이 그들을 머물도록 유인할 것인가?
	평생 가치	각 고객 계층의 평생 가치는 무엇인가?
	RCM(Revenue Cycle Management: 수익 주기 관리)	어떤 고객이 채무를 불이행 할 가능성이 있는가?
	매출	제품과 서비스의 예상 판매량/매출은? 어떻게 하면 영업 팀의 이직률을 줄일 수 있을까?
	CSat(Customer Satisfaction Score: 고객 만족 지수)과 NPS(Net Promoter Score: 순 추천 지수)	어떤 활동들이 CSat과 NPS에 가장 큰 영향을 미칠 것인가?
	고객 서비스	어떻게 일상적인 업무를 자동화해서 서비스 인력의 일상적인 작업을 줄일 수 있을까?

	프레임워크/문제	대표적인 비즈니스 질문
운영	RPA	복잡한 프로세스 특히, 비즈니스 규칙이 잘 정의되어 있지 않고 빈번하게 바뀌는 프로세스를 어떻게 자동화 할 것인가?
	PM(Preventive Maintenance: 예방 정비), 예를 들면, 장비와 네트워크	어떻게 하면 비용효과적으로 다운시간과 생산성 손실을 예방할 수 있을까?
	공급 체인	재고 관리, 생산 계획, 그리고 구매에 어떤 개선을 할 수 있을까?
	QA(Quality Assurance: 품질 보장)	어떤 제품에 품질 관련 문제가 있는가?
	운영 모델	고객 계층에 서비스를 제공하고 있는 개개 채널에 어떤 공유 서비스가 가장 효과적일까?
	ABC, LSS, RCA	여러 가지 프로세스 속성을 변경함으로써 폐기물과 품질 지표가 얼마나 개선될 수 있을까?
	사이버 분석, ITOA, 데브옵스(DevOps)	다량의 입력 신호와 경고 중에서, 어떤 것이 진짜 문제일까?

	프레임워크/문제	대표적인 비즈니스 질문
재무, 기업 전략, HR	프로젝트 평가: NPV, BE, ROI, 위험	더욱 연관성 있는 데이터 포인트를 활용하는 것처럼 추정치의 정확도를 높이는 방법
	사기 회피	사기를 회피하면서 오 탐지로 인한 고객의 불편은 최소화하는 방법
	핵심 역량	성공적인 결과 도출에는 어느 비즈니스 역량이 가장 중요한가?
	게임 이론	자사 전략 방안이나 업계 변화에 대한 경쟁사의 반응은 어떨 것 같은가?
	직원 몰입도	업무공간의 변화에 대한 직원의 반응은 어떨 것 같은가?
	채용	해당 지원자가 조직에 잘 맞을 가능성은?
	UBA	어떤 직원이 잘 못된 행동으로 기업에 해를 주거나 이직할 가능성이 있는가?

*14만~19만 명의 분석 기술 인력 공백이 존재한다.
그러나 관리자 수에 비해 효과적인 의사 결정을 내리기 위해 분석 사용 방법을 알고 있고, 적절한 질문을 할 수 있으며, 분석 결과를 효율적으로 사용할 수 있는 분석가의 공백은 150만명에 이르고 있다.

AI, 머신러닝, 딥 러닝은 명시적 프로그래밍과 전통적 통계 분석보다 뛰어나다

즈니스와 프로세스 전문가가 AI, 머신러닝, 딥 러닝을 잘 알고 있는 비즈니스 프레임워크와 개념에 접목할 수 있는 기회다. 이런 프레임워크 하에서 여러 가지 문제와 가설을 정의한 다음, AI, 머신러닝, 딥 러닝을 사용하여 패턴(통찰력)을 찾고, <표 2>처럼 실험 기간이 너무 길거나, 다른 방법으로는 찾아서 실험하기가 너무 비싸거나, 또는 인력으로 수행하기에는 너무 어려운 여러 가지 가설을 실험할 수 있다.

비즈니스가 점점 더 복잡해지면서 조직과 기업은 점점 더 AI, 머신러닝, 딥 러닝에 눈을 돌리고 있다. 우리 인간이 처리하기에는 한꺼번에 벌어지는 일이 너무 많은 것이다. 즉, 우리가 통합하기에는 너무 많은 데이터 포인트(관련이 있든 없든)가 있다. 너무 많은 데이터는 골칫거리가 될 수도 있는데 말이다.

그렇지만 AI, 머신러닝, 딥 러닝은 중요도를 제계적으로 판단하고, 산출물을 예측하며, 특정 사항을 미리 준비하고, 의사결정 과정을 자동화함으로써 이런 다량의 데이터를 자산으로 바꿔 놓을 수 있다. 요약하면 AI, 머신러닝, 딥 러닝은 조직과 기업의 복잡성을 주도하는 다음과 같은 요인을 다룰 수 있게 해준다.

- 더욱 글로벌해지고, 서로 얽혀있으며, 마이크로세그먼트에 집중된 가치 사슬과 공급 체인
- 경쟁업체 그리고 고객의 필요사항과 기호에 맞춰 빠르게 변화하는 비즈니스 규칙
- 경쟁 프로젝트/투자 그리고 비즈니스 지표를 최적화하기 위한 부족 자원의 정확한 예측과 배포
- 비용을 절감하면서 개선된 품질과 고객 경험을 동시에 추구해야 하는 필요성 여러 면에서, AI, 머신러닝, 딥 러닝은 명시적 프로그래밍과 전통적인 통계 분석보다 뛰어나다.
- 목표 결과를 달성하기 위해 비즈니스 규칙을 정말로 알 필요는 없다. 기기는 대표적인 입력과 출력만 학습되면 된다.
- 비즈니스 규칙이 변경돼서 동일한 입력이 더 이상 동일한 출력을 내놓지 않더라도, 기기를–재 프로그래밍이 아닌–재학습시키기만 하면 되어서, 응답 시간을 가속화하고 사람이 새로운 비즈니스 규칙을 배워야 하는 필요성을 줄여준다.
- 전통적인 통계 분석과 비교하여 AI, 머신러닝, 딥 러닝 모델은 상대적으로 빨리 구축할 수 있어서, 시도–학습–재 시도 접근방식에서 몇 가지 모델을 신속하게 반복할 수 있다.

그렇지만, AI, 머신러닝, 딥 러닝은 <표 3>과 같은 단점도 있다. 그런 단점 중 몇 가지는 여전히 통계에 기반을 두고 있어서, 산출물에 불확실성 요소가 존재한다. 기기의 의사결정에서의 높은 모호성을 사람이 처리해야 하기 때문에 AI, 머신러닝, 딥 러닝과 워크플로우의 통합을 까다롭게 만든다. 그리고 기기의 정확성을 개선하기 위해, 추가적인 훈련(학습)용으로 실수 사례(그리고 정답)을 기기로 재입력해야 한다.

또한, AI, 머신러닝, 딥 러닝 모델의 해석이 더 어려워질 수 있다. 즉, 의사결정에 도달하는 과정이 명확하지 않을 수 있다. 이는 여러 "계층(Layer)"과 뉴런(Neuron)을 가지고 있는 복잡한 딥 러닝 모델에서는 더욱 그렇다. 명확성 부재는 규제가 심한 산업에서는 특히 관심사가 된다. 현재는 이 분야에서 많은 연구가 이루어지고 있어서, 단점이 조금씩 줄어들 것으로 보인다.

AI, 머신러닝, 딥 러닝은 비즈니스 인텔리전스의 자연스러운 진전

장점과 단점을 감안할 때, AI, 머신러닝, 딥 러닝을 언제 활용하는 것이 좋을까? 몇 가지 견해를 소개한다.

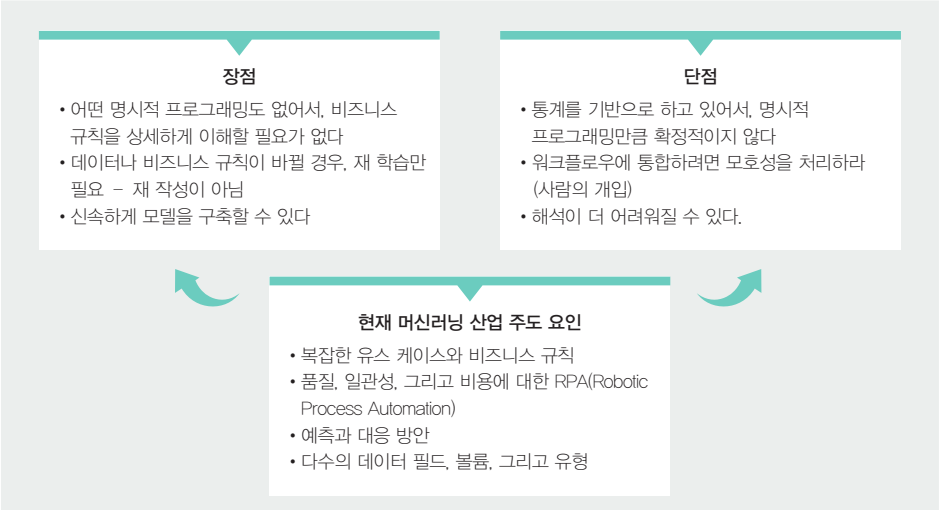
- 노력할 만한 가치가 있는 경우 : 비즈니스 성과에 대한 높은 잠재력이 있지만 전통적인 접근 방식은 너무 번거로우며, 시간이 많이 걸리거나, 그저 적합하지 않다.
- 관련 데이터를 사용할 수 있고 액세스할 수 있다.
- 주제 관련 전문가가 데이터에 의미 있는 신호가 있다고 믿고 있다(즉, 데이터로부터 통찰력을 얻을 수 있다).
- 문제 정의가 분류, 클러스터링, 또는 이상 탐지(Anomaly Detection)같은 머신러닝이나 딥 러닝 문제와 연계되어 있다.
- 유스 케이스(Use Case)의 성공을 정밀도 재현과 정확도 같은 머신러닝과 딥 러닝 모델 성능 지표에 매핑할 수 있다.

AI 정의 : BI에서 AI로의 자연스러운 진전

AI, 머신러닝, 딥 러닝은 비즈니스 인텔리전스의 자연스러운 발전이다. BI가 과거의 이벤트를 정의하고 진단한다면, AI, 머신러닝, 딥 러닝은 미래 이벤트의 가능성을 예측하고 그런 이벤트가 실제로 발생할 가능성을 높일 수 있는 방안을 수립한다. 이 과정을 설명하는 간단한 예가 포인트 A에서 포인트 B까지 길 안내를 하는 GPS다.

- **설명** : 차량이 어떤 경로를 택했으며, 시간이 얼마나 걸렸나?
- **진단** : 왜 특정 교통 신호에서 오랜 시간이 걸렸는가(GPS 플랫폼/도구가 사고와 교통량 등을 추적한다고 가정했을 때)?
- **예측** : 차량이 포인트 A에서 포인트 B로 갈 때, 예상 도착 시간은?
- **방안** : 차량이 포인트 A에서 포인트 B로 이동할 경우, 예상 도착 시간을 달성하기 위해서는 차량이 어느 경로를 택해야 할까?

표 3 | AI, 머신러닝, 그리고 딥 러닝의 장점, 단점 그리고 주도 요인



AI에서의 예측

예측 사례 하나를 들어 보자. 감성 분석(누군가가 어떤 것을 좋아할 확률)이다. 임의적 사용자의 게시물(트윗, 업데이트, 블로그 기사, 그리고 포럼 메시지 등)의 텍스트 콘텐트를 추적하고 저장할 수 있다고 가정해보자. 그렇다면 사용자의 게시물을 통해 해당 사용자의 감성을 예측하는 모델을 구축할 수 있다.

또 다른 예는 늘어나고 있는 고객 전환이다. 사람은 자신이 원하는 경품을 받을 수 있는 기회가 있을 때 가입 신청을 할 가능성이 더 높다. 어떤 경품이 가장 많은 수의 전환을 이끌어 낼지를 예측할 수 있을 것이다.

AI에서의 대응 방안

대응 방안의 핵심은 마케팅, 세일즈, 그리고 고객 서비스 같은 다양한 프로세스의 비즈니스 지표 최적화이며, 지시적 분석 시스템(Prescriptive Analytics System)에 최적화해야 하는 지표를 알려줌으로써 이를 달성할 수 있다. 최소 연료 소비, 가장 빠른 시간, 가장 짧은 거리, 또는 뭔가를 먹고 싶은 경우라면 가장 많은 수의 패스트푸드 편의점을 통과하는 조건 등 최적화하고 싶은 사항을 GPS에 알려주는 것과 같다. 비즈니스 환경에서는 10%의 전환 증가, 20%의 판매 증가, 또는 NPS 5포인트 증가를 목표로 할 수도 있다.

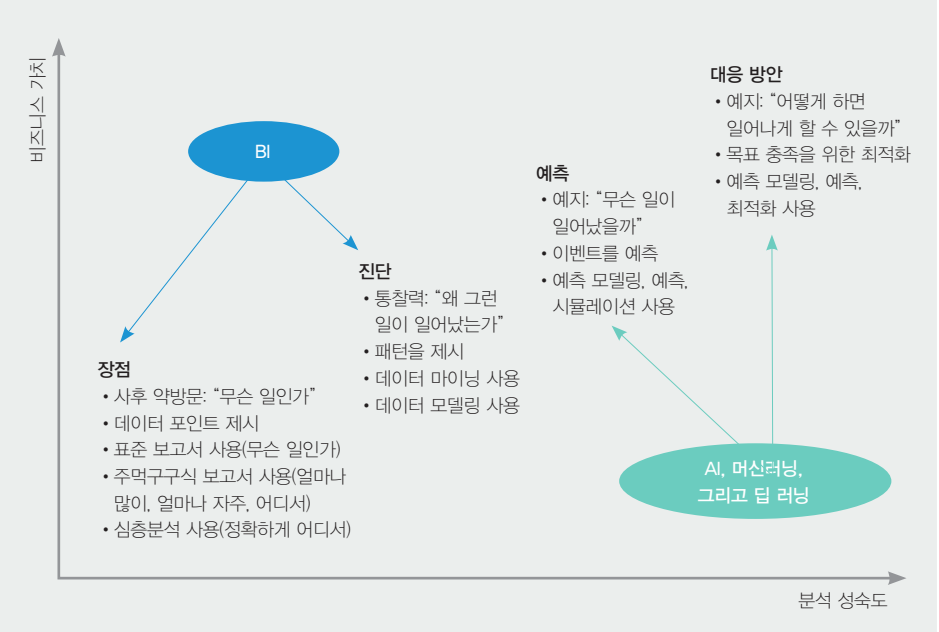
대응 방안의 핵심은 마케팅, 세일즈, 그리고 고객 서비스 같은 다양한 프로세스에 있어서의 비즈니스 지표 최적화

여기서 지시적 분석 시스템은 사용자가 원하는 사업 결과물로 이어지는 일련의 조치를 내릴 것이다.

가령 10%의 전환 증가율이 목표라고 하자. 시스템이 내릴 조치는 다음과 같다.

- DM(Direct Mail) 마케팅 빈도를 15% 줄이는 동시에,
- 트위터와 페이스북 참여를 동시에 각각 10%와 15%로 늘린 다음,

표 4 | AI, 머신러닝, 그리고 딥 러닝은 BI를 보완



• 전체 소셜 미디어 참여가 12%에 도달하는 시점에 일반 대중을 자사의 고객–대–고객 참여를 위한 고객 커뮤니티 포탈로 유도하기 시작

이런 대응 조치는 GPS 시스템이 사용자 설정 목표를 최적화할 때 이동 중에 지시하는 방향 전환과 같다.

BI, 통계, 그리고 AI 간의 관계

다음은 BI, 통계, 그리고 AI 간의 차이점이다.

• **BI(비즈니스 인텔리전스)**는 전통적으로 쿼리 지향적이며 패턴을 파악하기 위해 애널리스트 의존도가 크다(예를 들어, 수익이 가장 높은 고객은 누구인가, 그들의 수익이 가장 높은 이유는, 그리고 나이, 또는 직업 유형 같은 다른 고객과 차별화되는 속성은?).

• **통계**도 데이터에서 모집단에 대한 정보를 찾아내기 위한 데이터의 속성 이해를 위해 애널리스트에 의존하지만, 일반화에 대한 추정에 수학적 엄격함을 더한다(실제 생활에서 이런 고객 세그먼트와 표본 데이터에서 발견되는 것 간에 차이가 있는지).

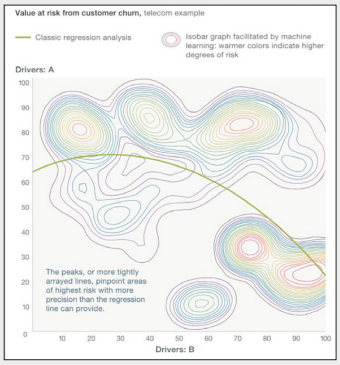
데이터에서 자율적으로 패턴을 찾고, 예측과 대응 방안을 활성화하기 위해 AI, 머신러닝, 딥 러닝은 애널리스트가 아닌 알고리즘에 의존한다

표 5 | BI, 통계 분석, 예측(Predictive) AI, 머신러닝, 그리고 딥 러닝

질문	분석 방법
수익이 가장 높은 고객은 누구인가	데이터베이스 쿼리
그 사람들의 수익이 가장 높은 이유는? 그들은 무엇이 다른가?	가족 구성원 변경, 직업, 거주지, 다른 구매 등 구매로 이어지게 할 수 있었던 상황에 대한 데이터베이스 쿼리
수익이 높은 고객과 일반 고객들간에는 정말로 차이점이 있는가?	통계 분석 • 가설의 긍정 또는 부정 • 차이점이 진짜일 것 같은 확률이나 신뢰도 도출
그렇다면, 대체 그 고객들은 누구인가? 그들을 특정할 수 있을까?	알고리즘이 수익이 있는 고객과 수익이 없는 고객들을 구분 짓는 특징을 결정한다
일부 특정 신규 고객은 수익이 높을까? 그 고객이 창출할 것으로 예상되는 매출은?	고객 기록에 대한 이력을 검토하고 신규 고객에 적용되는 매출과 수익성 예측 모델을 만들어내는 알고리즘

표 6 | 통계 모델링 Vs. 머신러닝

	통계 모델링	머신러닝
예측 근거	• 수학 방정식 형태의 변수 간 관계를 공식화 • 변수 의존도(Y)=f(독립 변수)+오차 함수	• 알고리즘을 활용해 데이터에 숨겨진 패턴을 찾으려고 하며, 데이터 메커니즘을 미지수로 취급 • 출력(Y) → 입력(X)
모델 구축 요구 사항	• 사전에 꼭 변수 간 관계를 이해해야 함 • 데이터에 대한 몇 가지 가정에 의존함 • 명시적 프로그래밍	• 사전에 변수 간 관계를 이해하지 않아도 됨 • 가정에 독립적이며, 그래서 일반적으로 예측 능력이 매우 뛰어남 • 명시적 프로그래밍 없이도 데이터를 학습



데이터의 속성이 바뀌더라도 머신러닝과 딥 러닝 모델을 새로운 데이터로 재교육하면 된다

- **AI, 머신러닝, 딥 러닝**은 자율적으로 데이터에서 패턴을 찾고 예측과 대응 방안을 활성화 하기 위해 애널리스트가 아닌 알고리즘에 의존한다.

BI, 통계, 그리고 AI, 머신러닝과 딥 러닝은 <표 5>에 설명된 것 이상을 할 수 있다. 이 예는 단지 이 방법이 일련의 사업적 질문에 어떻게 대답할 수 있는지를 보여주고 있을 뿐이다.

한쪽에는 통계 모델링이 있고 다른 쪽에는 머신러닝과 딥 러닝이 있다. 모두 사업 모델 구축에 사용되고는 있지만, 둘 간에는 <표 6>처럼 커다란 차이점이 있다. 특히,

- 통계 모델링은 입력과 출력 간에 반드시 공식적인 수학 방정식을 필요로 한다. 반면에, 머신러닝과 딥 러닝은 그런 수학 방정식을 찾으려 하지 않는다. 대신 단순히 입력이 주어지면 다시 출력 생성을 시도할 뿐이다.
- 통계 모델링은 변수 간 이해를 필요로 하고, 데이터 모집단의 통계적 속성을 가정한다. 머신러닝과 딥 러닝은 그렇지 않다.

보통 수학 방정식과 데이터 간의 관계를 이해해야 하기 때문에, 통계 모델은 통계학자가 데이터를 조사하고 작업을 해야 해서 구축에 비교적 긴 시간이 걸린다. 그렇지만, 성공적으로 완료되기만 하면–즉, 방정식을 찾아내고 데이터 간의 통계적 관계가 아주 잘 이해되었다면– 이 모델은 아주 훌륭한 결과물이 될 수 있다.

한편, 머신러닝과 딥 러닝 모델은 구축이 매우 빠르지만 처음부터 고성능을 내지는 못할 수도 있다. 그러나 초기 단계 구축이 아주 쉽기 때문에, 여러 개의 알고리즘을 동시에 시도해 보고 그 중 가장 가능성 있는 것을 골라 계속 반복해서 모델 성능을 최상으로 끌어올린다.

머신러닝과 딥 러닝 모델은 "스스로" 새로운 데이터를 가지고 끊임없이 학습한다는 추가 장점도 가지고 있다. 데이터의 속성이 바뀌더라도, 머신러닝과 딥 러닝 모델을 새로운 데이터로

표 7 | AI 대 머신러닝 Vs. 딥 러닝

2가지 유형의 AI가 있다:

AGI(Artificial General Intelligence: 범용 인공 지능)는 사람처럼 생각하고 사람과 비슷한 일을 하는 기계를 가리킨다. 아이 로봇 같은 영화에서 자주 보는 것이다.

ANI(Artificial Narrow Intelligence: 전용 인공 지능)는 특정 프로세스의 자동화 또는 해당 프로세스에서 특정 작업의 자동화처럼 매우 특정적인 작업에 기계를 사용하는 것을 말한다.

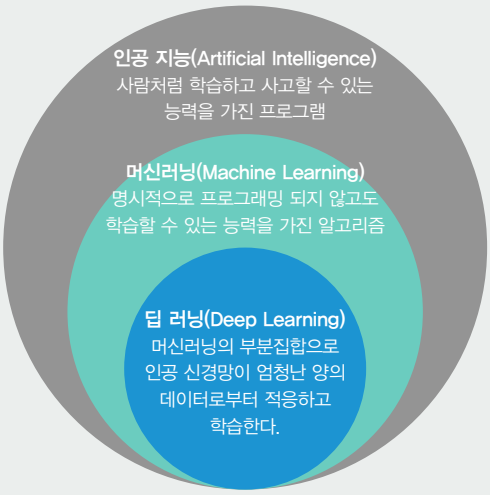
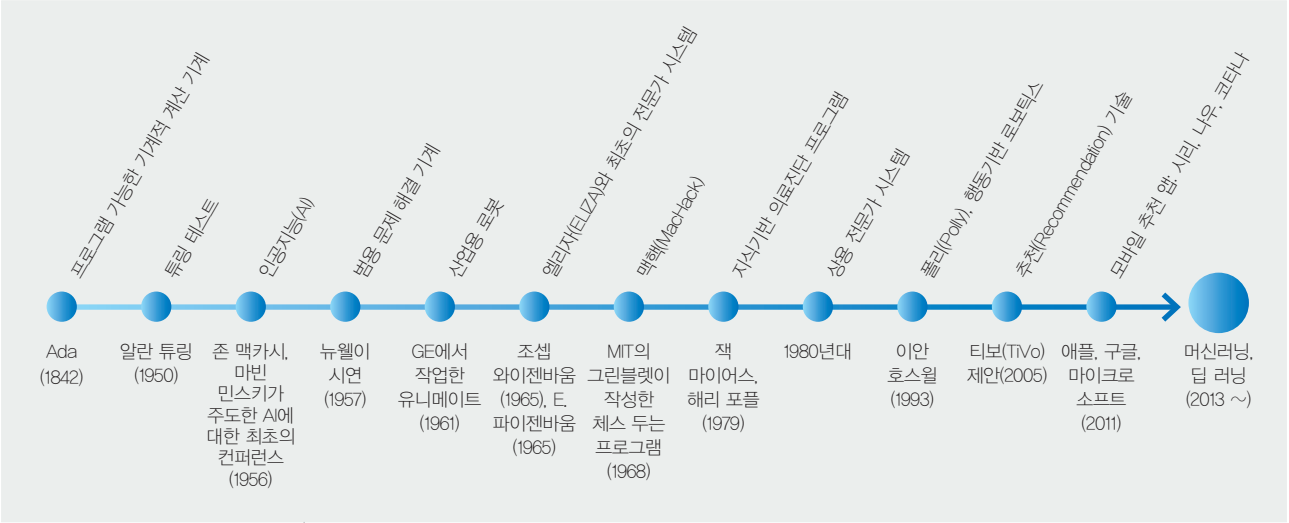


표 8 | AI 연대표



머신러닝이란, 명시적으로 프로그래밍되지 않은 기계가 알고리즘을 사용하여 작업을 학습하고 실행하는 것

재교육하기만 하면 된다. 반면, 통계 모델은 보통 전면적으로, 또는 부분적으로 재구축되어야 한다는 점이 다르다.

머신러닝과 딥 러닝 모델은 고도로 비선형적인 문제를 해결하는 데도 탁월하다(사람의 경우는 이 작업이 매우 어렵다–이런 방정식은 아주 길어질 수 있다!). 머신러닝과 딥 러닝의 이런 속성은 마이크로세그먼트가 표준(하나의 고객 세그먼트, 대규모 맞춤화, 개인화된 고객 경험, 그리고 개인별 정밀 의약품을 생각해 보라)이 되어 가고, 각종 프로세스와 근본 원인 분석이 갈수록 더 다원적이며 상호의존적으로 변해감에 따라 유용성은 더욱 커지고 있다.

AI, 머신러닝, 딥 러닝은 어떻게 다른가

지금까지 필자는 AI, 머신러닝, 딥 러닝을 하나로 뭉뚱그려 다뤘다. 그렇지만 <표 7>에서 알 수 있듯, 이 세 가지가 모두 같은 개념은 아니다.

일반적으로, **AI**는 기계가 인간 지능의 특징을 지닌 작업을 하는 것을 말한다. 이런 작업으로는 계획, 언어 이해, 객체와 소리 인식, 학습, 그리고 문제 해결이 있다. 이는 범용 인공지능(AGI)나 전용 인공지능(NGI) 형태가 될 수 있다.

- AGI는 사람의 모든 감각, 모든 추론 능력과 함께, 인간 지능의 모든 특징을 가지고 있어서 마치 사람처럼 생각할 수 있다. 일부에서는 이를 "코그니티브(Cognitive, 인지적 지능)"라고 설명하고 있다. 스타워즈에 나오는 C3PO를 떠올려보자.
- ANI는 인간 지능의 전체가 아닌 단 몇 가지 측면만 지니고 있다. 특정 작업을 수행할 때 사용된다. 예로는 핀터레스트에서의 이미지 분류, 페이스북에서의 안면 인식이 있다. ANI는 현재 비즈니스 애플리케이션에서 각광 받는 기술이다.

머신러닝은 기계가 명시적으로 프로그래밍 되지 않은 상태로 알고리즘을 사용하여 작업을 학습해서 실행하는 것을 말한다. 즉, 데이터를 통해서 학습하기 위해 특정 비즈니스 규칙을 제

공할 필요가 없다. 다른 말로 하면, "X가 보이면, Y를 실행해라"같은 명령어가 필요 없다.

딥 러닝은 머신러닝의 부분집합으로, 보통 인공 신경망(Artificial Neural Network: ANN)을 사용한다. 딥 러닝의 이점은 이론적으로는 어떤 데이터 속성(Data Element)이 중요한지를 알려줄 필요가 없다는 것이지만, 대부분의 경우 다량의 데이터가 필요하며, 이론적으로는 어떤 데이터 속성(또는 머신러닝 용어로는 "피처(Feature)")이 중요한지를 알려줄 필요가 없다.

<표 8>은 AI 진화 연대표를 보여준다.

명시적 프로그래밍, 머신러닝, 그리고 딥 러닝 간의 차이점은 필기 숫자 인식 예를 통해서 더 잘 알 수 있다. 5세 이상의 사람이라면, 필기 숫자를 인식하는 것이 어렵지 않다. 사람은 수년 간 부모, 선생님, 형제자매, 그리고 반 친구에 의해 학습을 받았기 때문이다.

이제 명시적 프로그래밍을 통해서 기계에 같은 일을 맡긴다고 상상해보자. 명시적 프로그래밍에서는 기계에게 무엇을 보아야 할지를 알려주어야 한다. 예를 들면, 둥근 물체는 0, 위에서 아래로 가는 줄은 1 등이다. 그러나, 물체가 완벽하게 둥글지 않거나, 끝이 이어지지 않아서 둥글지 않다면 어떻게 될까? 줄이 위에서 아래로 가는 대신 옆으로 기울어지거나 줄 맨 위에("1"처럼) 고리가 있다면-7에 더 가까운 모양이 아닐까?-어떻게 되는 걸까?

무엇을 찾아야 할지를 가르치는 것이 인간이기 때문에, 기계의 우수함은 그 식별 교육자 수준을 넘을 수 없다

<표 9>처럼, 머신러닝 접근방식에서는 사용자가 기계에게 여러 개의 1과 2 같은 예를 보여주고, 어떤 "피처"(중요한 특징)를 찾아야 할지를 알려준다. 모든 특징이 중요한 것은 아니기 때문에 이런 피처 엔지니어링이 중요하다. 중요한 특징의 예로는 폐쇄 루프 개수, 선의 개수, 선의 방향, 선의 교차점 수, 교차점의 위치를 들 수 있을 것이다. 중요하지 않은 특징의 예로는 색깔, 길이, 폭, 그리고 높이가 있다. 기계에 올바른 피처를 입력하고 여러 가지 예와 정답을 제공했다고 가정하면, 그 기계는 결국 서로 다른 숫자에 대해서 피처가 얼마나 중요한지를 스스로 학습하고, 여러 숫자를 정확하게 구별(또는 분류)할 수 있게 될 것이다.

머신러닝에서는 사용자가 기계에 중요한 특징(즉, 무엇을 찾아야 하는지)을 알려줘야 하기

표 9 | 명시적 프로그래밍 대 머신러닝과 딥 러닝: 필기 숫자 인식

	명시적 프로그래밍	머신러닝	딥 러닝
설명	• 특정 작업을 완수하기 위한 특정 명령어 세트를 가지고 있는 직접 작성한 소프트웨어 루틴	• 컴퓨터를 학습시켜서 명시적 명령어 없이 작업(예측)을 수행하기 위한 능력 획득	• 기계가 자동으로 피쳐를 학습하는 머신러닝 • 각 계층이 다음 계층으로 데이터를 전달하는, 여러 계층의 신경 세포(Neuron: 뉴런)를 가지고 있는 신경망 기반
예: 필기 숫자 인식하기	• 각각의 숫자를 구별할 수 있는 규칙 세트를 고안. 예를 들면, 0은 기본적으로 하나의 폐 루프다. • 잠재적 문제: 끝이 제대로 연결되지 않거나, 루프의 오른쪽 끝이 루프 왼쪽 끝이 시작하는 곳 아래에서 끝나서 6처럼 보이는 것 같은 불완전한 0 • 통과 기준 같은 규칙과 추측 목록을 작성하기가 복잡함	• 머신러닝은 새로운 상황에서 동일한 문제를 해결하기 위해 수 많은 사례로부터 학습하기 위한 (경험을 얻기 위한) 알고리즘을 개발 • 잠재적 문제: 알고리즘의 효율성이 유용한 피쳐를 찾아내기 위해 데이터 과학자의 능력에 의존한다	• 스스로 올바른 피쳐에 집중하는 것을 학습할 수 있는 딥 러닝 모델

때문에, 기계의 우수함은 그 식별 교육자 수준을 넘을 수 없다는 점을 기억해야 한다.

딥 러닝은 기계에게 어떤 피처를 사용할지(즉, 어떤 것이 가장 중요한지)를 아무도 알려주지 않아도 된다고 장담하는 기술이다. 기계가 자동으로 알아낸다는 것이다. 사용자는 기계가 스스로 중요한 피처를 선정하게 될 모든 피처를 입력해주기만 하면 된다. 명확한 장점이기는 하지만, 고용량의 데이터 요구사항과 많은 계산 처리 용량을 필요로 하는 긴 학습 시간이라는 대가를 치러야만 한다.

AI 모델 개념 : 개요

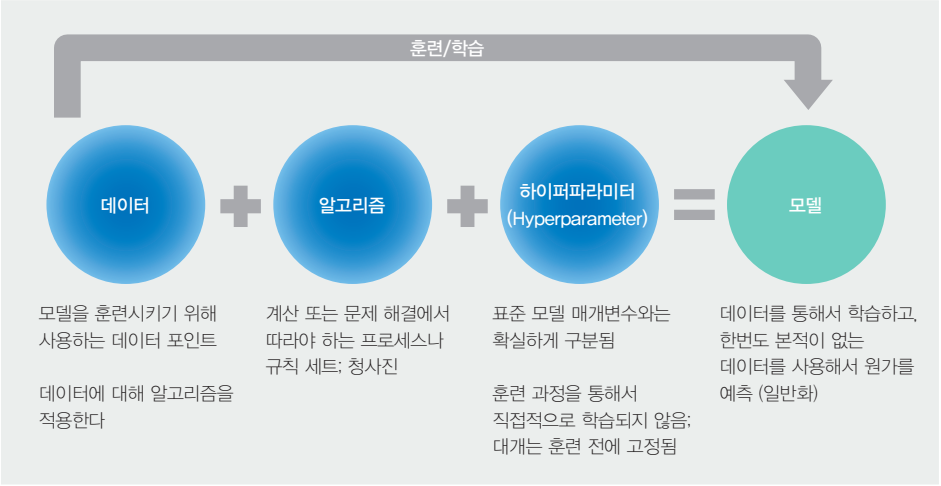
머신러닝과 딥 러닝 모델에는 주어진 데이터(전에 보았던 것)를 가지고 학습한 다음, 새로운 데이터(전에 본 적이 없는 것)에 대해 올바른 의사결정을 하기 위한 일반화를 하자는 생각이 깔려있다.

그런다면 모델은 무엇으로 구성되는가? 모델이 3가지 구성요소로 이루어진다는 점이 한 가지 정의가 되겠다.

이론적으로는 훈련 수준에 따라 처음 보는 악보도 연주할 수 있다

- **데이터**: 모델을 학습할 때 과거 데이터(Historical Date)를 사용한다. 예를 들어, 피아노 연주를 학습할 때 입력하는 데이터로는 여러 가지 음표, 상이한 유형의 음악, 서로 다른 작곡가 스타일 등이 있다.
- **알고리즘**: 학습 프로세스에서 모델이 사용하는 일반 규칙. 다시 피아노를 예로 들면, 내부 알고리즘이 악보, 피아노 운지법, 언제 그리고 어떻게 페달을 밟아야 하는지 등을 알아보라고 말해 줄 수도 있다. 〈표 10〉은 모델과 알고리즘 간의 관계를 보여준다.
- **하이퍼파라미터**: 데이터 과학자가 모델 성능을 개선하기 위해 조절하는 "손잡이" 역할을 한다. 데이터에서 학습되지 않는다. 피아노 연주를 예로 들면, 하이퍼파라미터는 음악 작품을 얼마나 자주 연습하는지, 어디서 연습하는지, 연습하는 시간, 연습용으로 사용하는 피아노가 무엇인지 등을 포함한다. 이런 "손잡이"를 조절하면 피아노 치는 법을 학습하는 능력이 개선된다는 생각이다.

표 10 | 모델과 알고리즘 간의 관계



훈련 데이터는 일련의 예측 변수(독립 변수)로 예측될 목표/결과 변수(또는 종속 변수)로 구성된다

이 모든 것을 합치면 피아노 연주 모델이 된다. 이론적으로는 얼마나 잘 훈련 받았느냐에 따라, 처음 보는 새로운 음악의 악보라고 해도 그 연주 여부가 결정될 것이다.

머신러닝의 유형

<표 11>처럼 기계도, 마치 사람처럼, 여러 가지 방법으로 학습할 수 있다. 다시 피아노 훈련 비유를 사용해서 설명해보자:

- **지도(Supervised)** : 교사가 올바른 방식과 잘못된 방식 2가지 모두를 알려주거나 보여준다. 이상적인 상황에서는, 어떻게 하면 옳고 그른 방법을 수행할 수 있는지에 대한 예가 동등하게 주어진다. 기본적으로, 훈련 데이터는 일련의 예측 변수(독립 변수)로 예측될 목표/결과 변수(또는 종속 변수)로 구성된다. 이런 변수 세트를 사용해서, 입력을 원하는 출력으로 매핑하는 함수(Function)를 생성한다. 모델이 훈련 데이터에 대해 원하는 수준의 성능에 도달할 때까지 훈련이 계속된다. 지도 훈련을 사업에 접목한 예로는 승인되었거나 거부되었던(목표 결과/의사결정) 대출 신청 시스템(신용 기록, 직장 기록, 보유 자산, 수입, 그리고 학력 같은 예측 변수로 이루어진) 사례를 들 수 있다.
- **비지도(Unsupervised)** : 혼자 알아서 해야 한다. 누구도 방법을 알려주지 않기 때문에, 작품 완주 속도, 큰 소리와 부드러운 소리의 비율, 또는 누르는 고유 건반 개수 같이 중요한 매개변수를 최적화한다는 목표 하에, 옳고 그름을 스스로 판단해야 한다. 근본적으로, 데이터 포인트에는 옳거나 그름을 알려주는 어떤 라벨도 붙어있지 않다. 대신에, 목표는 데이터를 특정 방식으로 체계화하거나 데이터의 구조를 정의하는 것이다. 즉, 데이터를 클러스터(Cluster)로 그룹핑(Grouping)하거나 복잡한 데이터를 바라볼 때의 새로운 방법을 찾아내서 더 간단하거나 또는 더 체계적으로 만드는 작업이다. 모델 훈련에서 대개 비지도 학습이 지도 학습에 비해 효과가 적기는 하지만, 라벨(Label)이 존재하지 않는 경우에는 필요할 수 있다(다시 말

표 11 | 모델링: 공통적인 학습/훈련 유형

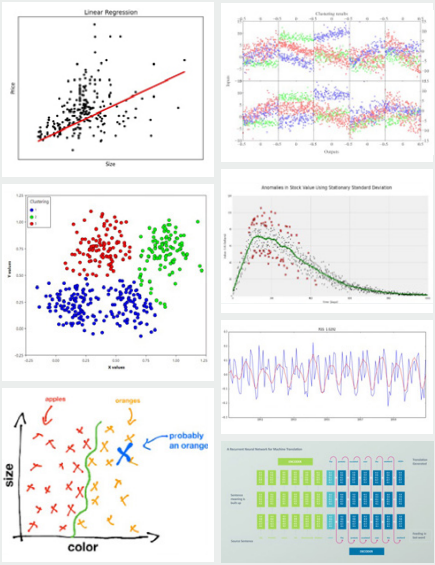
유형	설명	예
지도(Supervised)	• 원하는 출력이 알려져 있음(라벨 있는 데이터: Labeled Data – 목표 값이 표시된 데이터) • 입력이 적용되면서, 출력이 목표 값과 비교된다. 모델 출력을 목표 값에 더 근접시키기 위해 조정이 이루어진다	• 가입 취소 예측 • 필기체 인식 • 일기 예보 • 스팸 필터링
비지도(Unsupervised)	• 원하는 출력이 알려져 있지 않지만, 데이터들 간의 관계는 존재하는 것으로 여겨진다 • 모델은 데이터에서 근원적인 패턴/관계(유사점과 차이점)를 찾는다 • 모델이 관계 (데이터 카테고리)를 찾은 뒤에는 관계가 유용한지/실행가능한지를 판단하기 위해 추가 조사가 필요하다	• 세분화 • 추천 시스템 • 이상(특이점) 탐지
준지도(Semisupervised)	• 훈련을 위해 라벨 없는 데이터(Unlabeled Data)를 사용하는 지도 학습 작업과 기법 클래스 – 대개는 다량의 라벨 없는 데이터를 가지고 있는 소량의 라벨 있는 데이터	• 라벨 있는 예는 소규모지만 라벨 없는 예는 대규모 세트를 가지고 있는 문제 • 음성 분석, 단백질 서열 분류, 웹 콘텐트 분류
강화(Reinforcement)	• 목표를 달성하거나 여러 가지 단계에 대한 차원을 극대화한다: 잘못된 의사결정을 하면 벌칙을 받고 올바른 판단을 하면 보상을 받는다 • 가능한 결과 수가 너무 많아서 (체스에서 체스 수가 수만 가지인 경우처럼) 지도 학습을 사용하는 것이 비현실적일 경우 유용하다	• 포커, 주사위 놀이, 오델로, 체스 그리고 바둑 같은 논리 게임 • 무인 자동차, 자기 탐색 진공 청소기, 그리고 엘리베이터 순서 배치 같은 제어 문제
전이(Transfer)	• 한 작업을 위해 개발된 모델이 두 번째 작업을 위한 모델의 출발점으로 재사용되는 것(용도 변경 또는 학습된 피쳐를 전이) • 대개는 딥 러닝 모델에 사용됨	• 이미지와 언어

해서, "올바른" 답이 알려지지 않은 경우). 보통 사업에 활용되는 사례는 시장 세분화다. "딱 맞는" 시장 부문이 무엇인지 명확하지 않을 때가 있기 마련이다. 모든 마케터는 올바른 메시지, 판촉활동, 그리고 상품만으로 그런 부문에 접근할 수 있는 자연스러운 방식을 찾고 있다.

- **준지도(Semisupervised)** : 지도와 비지도의 결합이다. 지도 데이터가 충분하지 않을 경우에 사용된다. 피아노 예에서는 많지 않으나 약간의 지도를 받을 수도 있다(레슨 비용이 비싸거나 강사 수가 충분치 않을 경우).
- **강화(Reinforcement)** : 어떤 것이 옳고 그른 연주 방법인지 알려주지 않으며, 어떤 매개변수를 최적화하려고 하는지도 알 수 없다. 다만 뭔가를 제대로 했을 때, 또 제대로 하지 못했을 때 알려준다. 피아노 훈련의 경우, 잘못된 음표를 연주하거나 잘못된 속도로 연주하면 선생님이 손가락 마디를 자로 때리기도 하고, 연주를 잘 하면 등을 쓸어준다. 강화 학습은 현재 아주 인기가 높은 항목인데, 그 이유는 모든 시나리오에서 사용할 수 있는 지도 데이터가 충분하지 않지만, "올바른" 답은 알려져 있는 경우가 종종 있기 때문이다. 예를 들면, 체스 게임에서 기록할 체스 수의 순열(라벨)이 너무 많은 경우다. 그럼에도, 강화 학습은 기계에게 체스 말 획득과 체스 판에서 위치 강화처럼 승리로 이끌어가는 옳고 그른 의사결정을 내릴 때 알려준다.
- **전이 학습(Transfer Learning)** : 전이 가능한 어떤 기술(악보를 읽는 능력, 또는 더 나가서 양손의 민첩성 등)을 익혔다면 트럼펫 등 다른 악기를 배울 때 피아노 연주 지식을 사용한다. 학습 시간이 줄어든다는 점은 딥 러닝 아키텍처 사용 모델에 있어서 상당히 큰 강점이므로(몇 시간 심지어는 며칠) 전이 학습이 활용된다.

표 12 | 모델링: 알고리즘의 유형

알고리즘	설명	학습 유형	예
회귀(Regression) (2, 3, 4, 5)	데이터 포인트에 곡선/직선을 일치시켜서, 데이터 포인트 간의 거리 격차가 곡선 또는 직선으로부터 최소화 되도록 한다	지도	• 예를 들어 위치, 평 수, 또는 침실 개수에 근거한 집 값 • 주제, 날짜/시간, 또는 경품에 기초한 밋업 참석자
분류 (0/1/2/……/N)	새로운 관찰값(Observation)이 어느 카테고리 세트(하위 모집단)에 속하는 지를 규명	지도	• 고객 이탈 또는 비 이탈 • 사기 또는 정상 거래 • 질병 보유 또는 미 보유 환자; 질병 종류
클러스터링 (Clustering) (그룹핑)	데이터를 가장 공통점이 많은 그룹으로 가장 잘 체계화하기 위해 데이터에 내재된 구조를 사용. 그 어떤 오차(error) 또는 보상(Reward) 신호도 없음	비지도	• 시장 세분화, 예를 들면, 타깃 마케팅, 이탈율 감소 • 이상점 규명, 예를 들면, 고위험군 환자, 의심스러운 거래
시계열 (2010, 2011)	일정 기간 동안 수집한 데이터 포인트에는 내부 구조가 있을 수 있다 (예를 들면, 자기상관, 트렌드 또는 계절적 변화)	지도 또는 비지도	• 주식의 일간 종가 • 발작을 나타내는 EGG 추적 분석 • 서버의 시간 별 활용 수요
최적화	주어진 제약조건을 최대 또는 최소화	지도 또는 강화	• 기계 활용 최대화 • 수송 시간 최소화 • 가장 가치 있는 물건의 배달
NLP(자연어 처리)	인간 언어의 자동 계산 처리; 텍스트를 입력과 출력으로 사용	지도 또는 비지도	• 악보 번역 • 자동 완성, 다음 단어 제안 • 맞춤법 검사기
이상 탐지	이상점 탐지, 예상된 행태와 일치하지 않는 이상한 패턴 규명	비지도	• 장애가 예상되는 기계 • 침입을 시도하는 사람과 기계 • 참 대 거짓 경보



전이 학습의 장점은 시간 절약이며, 딥 러닝 아키텍처를 사용하는 모델에 꼭 필요한 요소다

일반적인 머신러닝 알고리즘

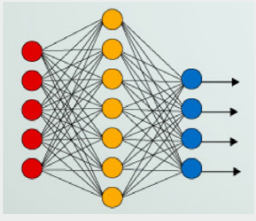
<표 12>에서처럼 일반적인 알고리즘 유형은 다음과 같다.

- **회귀**는 여러 데이터 포인트를 통과하는 곡선이나 직선을 그리는 것이다.
- **분류**는 어떤 대상이 속하는 그룹을 결정한다. 이진 분류(Binary Classification)(2개의 그룹)는 어떤 대상이 예를 들어, 그림 속의 동물이 개인지 아닌지처럼 특정 클래스에 속하는지 아닌지를 결정한다. 동물의 예를 다시 들면, 다중 분류(Multiclass Classification)(2개 이상의 그룹)는 그 동물이 개, 고양이, 새 등에서 어디에 속하는가를 말한다.
- **클러스터링**은 분류와 비슷하지만, 사전에 범주를 알 수 없다. 다시 동물 그림의 예를 사용하면, 3가지 유형의 동물이 있다고 정할 수 있지만, 그 동물이 무엇인지는 모르고 있기 때문에, 그냥 그룹별로 나누는 것만 할 수 있다. 보통 클러스터링은 지시 데이터가 부족하거나 개나 고양이 또는 새 같은 특정 그룹에 제약 받지 않고 데이터에 내재되어 있는 자연스러운 그룹을 찾고 싶을 때 사용한다.
- **시계열**은 데이터 순서가 중요하다고 가정한다(일정 기간 동안 수집한 데이터 포인트에 고려해야 할 내부 구조가 있다). 예를 들면, 계절성을 알아내기 위해 일정 기간 동안의 매출 추이를 파악해서 홍보 이벤트와의 연관성을 찾고자 할 수 있기 때문에 판매 데이터를 시계열로 간주할 수 있다. 반면에, 동물 그림의 순서는 분류 목적상 별다른 의미가 없다.
- **최적화**는 여러 개의 변수가 같은 방향으로 이동하지 않을 때, 최상의 가치를 달성하기 위한 방법이다.
- **NLP**(Natural Language Processing)는 챗봇, 주요 데이터 필드에 대한 의사의 노트 같은 비정형 수기 기록의 정리, 그리고 뉴스 기사 자동 작성 같은 인간의 언어 사용 능력과 이해를 흉내 내려 하는 일반 범주의 알고리즘이다.
- **이상 탐지**는 데이터에서 이상한 점을 찾아낼 때 사용된다. 공정 관리도와 비슷하지만 입력으로 훨씬 더 많은 변수를 사용한다. 이상 탐지는 "정상적인" 운영 매개변수를 정의하기 어

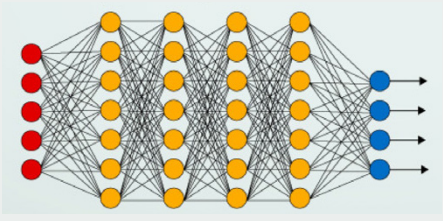
표 13 | 모델링: 딥 러닝

- 인공 신경망(Artificial Neural Network: ANN) 기반
- 아무런 피쳐 엔지니어링도 필요 없음
- 엄청난 양의 지도 훈련 데이터를 필요로 한다; 그렇지만, 모델 정확도는 데이터에 따라 변경된다
- 일반적인 예: 이미지 인식, NLP/음성

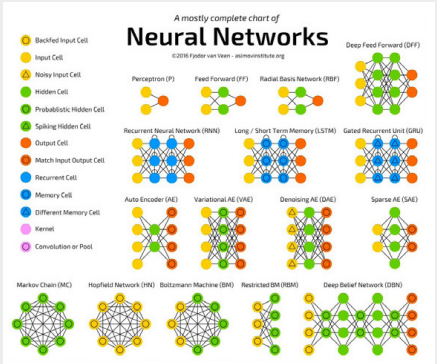
간단한 신경망
(Simple Neural Network)



딥 러닝 신경망



● 입력 계층 ● 숨겨진 계층 ● 출력 계층



딥 러닝은 인공 신경망(ANN) 개념에 기초한다. 인공 신경망은 피드백 종류에 따라 강해지고 약해지는 인간의 뇌처럼 작동하고, 신경 세포는 특정 조건에 따라 전기적 신호를 발생한다

렵고 시간이 흐름에 따라 바뀌는 경우에 특히 유용하며, 이상 탐지가 자동으로 조정되기를 바랄 것이다.

딥 러닝 모델

딥 러닝은 인공 신경망(ANN) 개념에 기초하고 있다. 그래서, 인공 신경망은 피드백 종류에 따라 강해지고 약해지는 인간의 뇌처럼 작동하고, 신경 세포는 특정 조건에 따라 전기적 신호를 발생한다. 자율 주행 자동차, 이미지 검출, 영상 분석, 언어 처리 같은 어려운 문제가 딥 러닝 모델이 해결하고 있다. <표 13>은 딥 러닝의 주요 특성을 보여준다.

딥 러닝 모델만을 사용해야 한다고 생각하지는 말자. 몇 가지 유의사항이 있다.

- 첫째, 많은 양의 데이터를 필요로 한다 – 일반적으로 머신러닝 모델보다 훨씬 더 많은 양. 다량의 데이터가 없다면, 딥 러닝은 보통 그리 잘 수행되지 않는다.
- 둘째, 딥 러닝이 다량의 데이터를 필요로 하기 때문에, 훈련 기간이 길고 많은 계산 처리 능력을 필요로 한다. 이 문제는 GPU와 FPGA(Field Programmable Gate Array)뿐 아니라 전에 없이 강력하고 빨라진 CPU와 메모리로 해소되고 있다.
- 셋째, 딥 러닝 모델은 대개 머신러닝 모델에 비해 해석이 어렵다. 해석능력은 딥 러닝 연구의 주요 분야이므로, 개선될 것으로 보인다.

머신러닝 모델 성능 측정 방법

모델도 사람처럼 성능을 평가받는다. 다음은 비교적 간단한 회귀 모델의 성능을 측정하는 몇 가지 방법이다. MAE, RMSE, 그리고 R2 성능 지수는 <표 14>에서처럼 비교적 복잡하지 않다.

이 모든 것을 비용 함수의 유형이라고 간주할 수 있으며, 이는 모델이 "올바른" 답에 점점 더 다가가고 있는지, 점점 더 멀어지는지, 그리고 답에 "충분히 가까워졌는지"를 아는 데 도움이 된다. 비용 함수는 과거에 없었던 새로운 데이터를 받기 위해 얼마나 더 가야 하는지를 모델에 알려주고 충분한 확률로 올바른 예측을 출력해낸다. 모델을 훈련할 때의 목표는 비용 함수를 최소화하는 것이다.

표 14 | 모델링: 성능 평가: 회귀에 대한 오차 계산 예

평균 절대 오차(Mean Absolute Error: MAE)
– fi가 예측 값이고 yi가 참 값인 경우, 절대 오차의 평균

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| = \frac{1}{n}\sum_{i=1}^{n}|e_i|.$$

RMSE(Root Mean Square Error: 평균 제곱근 오차)
- 모든 오차의 제곱에 대한 평균 값의 제곱근
- 작은 오차보다 큰 오차에 벌점을 준다

$$RMSErrors = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

R2 결정 계수(Coefficient of Determination)
- 모델이 데이터를 얼마나 잘 설명하는가
- 독립 변수에서 예측할 수 있는 종속 변수의 분산 비율
  – R2 값 0은 독립 변수에서 종속 변수를 예측할 수 없음을 의미한다.
  – R2 값 1은 독립 변수에서 오차 없이 종속 변수를 예측할 수 있음을 의미한다.

$$R^2 \equiv 1 - \frac{SS_{\mathrm{res}}}{SS_{\mathrm{tot}}}, \quad SS_{\mathrm{reg}} = \sum_i (f_i - \bar{y})^2, \quad SS_{\mathrm{tot}} = \sum_i (y_i - \bar{y})^2,$$

비용 함수(Cost Function)

$$J(\theta) = \frac{1}{2n}\sum_{i=1}^{n}(h_\theta(x^i) - y^i)^2$$

현실에서 정밀도와 재현율은 상충관계이므로 하나의 지표가 개선되면 다른 지표가 악화된다

표 15 | 모델링: 성능 평가, 분류에 대한 혼동 행렬

혼동 행렬

예측된 클래스

실제 클래스(Class)	클래스 = 예	클래스 = 아니오
클래스 = 예	TP(참 양성)	FP (거짓 음성: False Negative)
클래스 – 아니오	FP(거짓 양성)	TP (참 음성: True Negative)

참 양성 비(TPR: True Positive Rate) 또는 재현율(Recall) ←

거짓 양성 비(FPR: False Positive Rate) ←

↑ 정밀도

지표(Metrics) 사용

클래스 분포

비용(Cost)	균등	비 균등
FN(거짓 음성: False Negative)이 더 많은 비용이 듦	재현율	재현율
동일한 비용	정확도 또는 F1 스코어 (Score)	F1 스코어
FP가 더 많은 비용이 듦	정확도	정확도

정의

지표	방정식	해석
정확도	(TP + TN)/(TP + TN + FP + FN)	• 참 양성과 음성을 구분하기 위한 능력("정확")
참 양성 비 또는 재현율	TP/(TP + FN)	• 관련된 모든 인스턴스를 검색하기 위한 능력(완결도)
정밀도	TP/(TP + FP)	• 관련된 인스턴스만을 검색하기 위한 능력(관련도)
거짓 양성 비	FP/(FP + TN)	• 무관한 모든 인스턴스를 검색하기 위한 능력
F1 스코어	2 * 재현율 * 정밀도/ (재현율 + 정밀도)	• 정밀도와 재현율의 조화 평균 • 완결도와 관련된 것만을 가져오는 작업을 극대화시키는 드레이드오프(Trade-off)

분류 모델에서 정밀도 vs. 재현율

비용 함수가 훈련 데이터(보이는 데이터)를 기반으로 모델이 "올바른 답"이 있는 방향으로 가는데 도움을 주는 의무를 다한 다음에는, 아직 보지 못한 데이터에 대해 모델이 얼마나 잘 수행하고 있는 지를 평가해야 한다. 분류 모델(그림이 개, 고양이, 쥐 등등 인지의 여부 같이, 어떤 대상이 한 그룹에 속하는지 아니면 다른 그룹에 속하는 지를 결정하는 모델)의 맥락에서 설명해보자.

분류 모델의 성능을 평가하기 위해서(<표 15> 참조), 정확도(아래 자세히 설명) 방정식을 사용한다. 그렇지만, 훈련 데이터가 클래스 불균형을 보이는 경우, 정확도 지표(Accuracy Metric)가 판단을 그르칠 할 수도 있다는 것이 일반적으로 받아들여지고 있으므로, 대신에 정밀도와 재현율이라는 지표를 사용한다. 이런 용어의 의미는 다음과 같다.

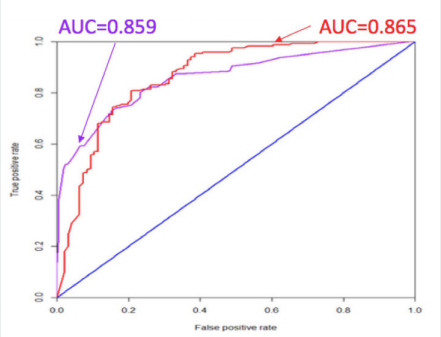
- **클래스 불균형**(Class Imbalance). 데이터가 이 방향 또는 다른 방향으로 치우친다. 신용카드 거래가 사기인지 여부를 예측하는 예를 생각해 보자. 대다수의 거래는 사기가 아니므로, 데이터는 그 방향으로 치우칠 것이다. 그래서, 특정 거래가 사기가 아니라고 예측했다면, 아마

도 맞을 것이다. 거래 자체에 대해 아는 것이 전혀 없더라도, 이 예에 정확도 지표를 적용하면 마치 사기가 아닌 거래를 예측하는 훌륭한 일을 하고 있는 것처럼 사용자를 호도할 수도 있다.

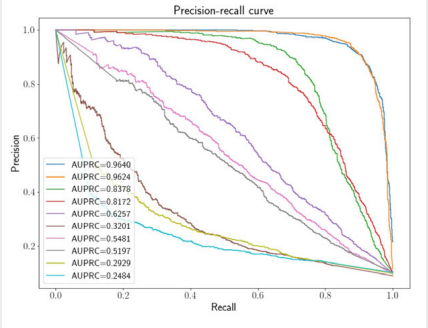
- **정밀도**는 관련도(Relevance)의 척도이다. 테니스 점수 "러브(Love)"의 기원을 찾기 위해 검색 엔진을 사용한다고 해보자. 정밀도는 검색 결과 항목 중 몇 개가 정말로 이 질문에 대한 것인지 아니면 테니스를 얼마나 사랑하는지, 사람들이 테니스를 치다가 어떻게 사랑에 빠지는지 등에 대한 링크인지를 판단한다.
- **재현율**은 완전도의 척도이다. 동일한 예인 테니스 점수 "러브"를 사용하면, 재현율은 검색 엔진이 사용할 수 있는 모든 참조사항을 얼마나 잘 수집했는지를 판단한다. 놓친 것이 전혀 없다면 놀라운 것이고, 한두 개를 놓쳤더라고 나쁘지 않지만, 수천 개를 놓쳤다면 끔찍한 것이다.

(속이지 않는 한 얻을 수 없는)완벽한 커브는 1을 향해 Y축으로 올라간 다음 최상단을 가로질러 가는 모양이다

표 16 | 모델링: 성능 평가: ROC커브와 PR커브



ROC(Receiver Operating Characteristic:
수신자 조작 특성 또는 수용자 반응 특성):
참 양성 비: TP/(TP + FB)
거짓 양성 비: FP/(FP + TN)



PR(Precision Recall: 정밀도 재현율):
재현율(완전도): TP/(FP + FN)
정밀도(관련도): TP/(TP + FP)

표 17 | 모델링: 저조한 성능으로 이어지는 일반적인 장애물

- **문제 간소화(Problem Formulation)**
  - 시험을 하기 위해 가설 설정
- **데이터 문제**
  - 가설과 연계된 데이터 세트가 전혀 없음
  - (지도 학습을 위한) 데이터와 라벨드(Labeled) 데이터 부족
  - 데이터의 의미에 대한 오해
  - 데이터에 어떤 신호도 없음
  - 값이 누락됨
  - 정규화(Normalization)
  - 클래스 불균형
  - 개념 드리프트(Concept Drift, 데이터의 통계적 변화(평균, 분포 등)가 지속적으로 발생하여, 예측하는 모델의 정확도가 시간이 지날수록 점점 낮아지는 현상)
- **적합한 모델 알고리즘과 아키텍처 선택**
  - 목적에 적합한 알고리즘과 아키텍처
  - 복잡성, 정확도 대 훈련 기간
- **올바른 피처 선택**
  - 더 많은 수의 피처: 전황도 대 훈련 시간
- **하이퍼파라미터 조정**
  - 순열/조합
- **모델 훈련**
  - 데이터 분할(Splitting), 검증, 성능 평가, 에포크(Epoch: 학습 반복) 횟수
- **비용(오차) 함수**
  - 오차 함수 선택
  - 전역(Global) 대비 로컬 최소점(Minima) 찾기, 그리고 최소점 찾기 속도
- **과소적합(Underfitting) (편향: Bias)**

데이터 과학 역시
예술이면서 과학이다

안타깝게도, 현실 세계에서, 정밀도(Precision)와 재현율(Recall)은 상충관계에 있다. 즉, 하나의 지표가 개선되면, 다른 지표가 악화된다. 그러므로, 어느 지표가 더 중요한 것인지를 결정해야만 한다.

어울리는 사람과 연결시켜주는 데이팅 앱을 생각해보자. 잘 생겼고, 부유하며, 뛰어난 개성을 지니고 있다면, 잠재적 상대가 매우 많을 거라는 점을 알고 있기 때문에 더 높은 정밀도 쪽으로 기울지도 모르지만, 정말 잘 어울리는 상대만을 원하며, 잠재적 상대를 가려내기 위한 비용이 높다. 반면에, 오랫동안 누군가를 찾고 있었고 부모가 압박을 가해오고 있다면, 가능한 많은 잠재적 상대를 얻기 위해 재현율 쪽으로 기울 수도 있다.

어울릴만한 상대를 자세히 살펴보는 비용은 비교적 낮다! 모델이 정밀도와 재현율 사이에서 균형을 잡고 있는지를 평가하기 위해 F1 스코어를 사용한다.

<표 16>처럼 이런 지표를 그래프 상에 표현할 수 있다. 그 중 하나를 ROC 커브(수신자 조작 특성 커브) 그리고 다른 하나를 PR 커브(정밀도-재현율 커브)라 부른다. 완벽한 커브(속이지 않는 한 얻을 수 없는!)란 1을 향해 Y축으로 올라간 다음 최 상단을 가로질러 가는 커브다. ROC 커브의 경우, 대각선을 가로지르는 직선은 나쁘다. 이는 모델이 참 양성과 참 음성을 각기 50%의 비율로 동일하게 예측하고 있음을 의미한다. 이런 지표는 종종 AUC(Area Under the Curve, 커브 아래 영역)으로 변환되기 때문에, AUC ROC 그리고 AUC PR 같은 용어를 보게 될 것이다.

머신러닝 모델 구축이 어려울 수 있는 이유

이제 모델이 무엇인지 그리고 모델의 성능을 어떻게 판단할 것인지를 이해했으니, 성능 좋은 모델을 구축하는 것이 왜 어려울 수 있는지를 자세히 알아보자. <표 17>처럼 몇 가지 이유가 있다. 그 중에서 문제 간소화, 데이터 문제, 적합한 모델 알고리즘과 아키텍처 선택, 올바른 피처 선택, 하이퍼파라미터 조정, 모델 훈련, 비용(오차) 함수, 그리고 과소적합(편향)과 과적합(Overfitting)(분산).

데이터 과학도 다른 과학과 마찬가지로 예술이자 과학이라는 것을 명심하라. 물론, 일을 강압적으로(Brute-force)하는 방법도 있지만, 그런 접근방식은 시간이 많이 걸리고, 통찰력을 잃을 수도 있으며, 자칫 일을 그르칠 수도 있다. 데이터 과학에서 통용되는 접근방식은 비즈니스 필요사항을 충족시키는 모델을 창출하기 위해(사업, 운영, 그리고 트랜스포메이션과 개선 전문가 같은) 해당 주제 전문가와 데이터 과학자의 중지를 모으는 것이다.

과적합 vs. 과소적합

과적합과 과소적합은 특히 인기 있는 문제 결과이므로, 조금 더 자세히 보기로 하자. <표 18>처럼 여기에는 편향과 분산이 개입된다.

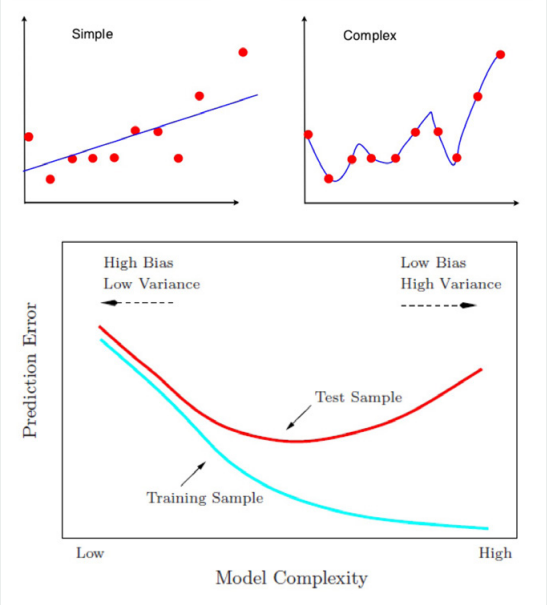
과적합(고분산)은 모델이 데이터의 변동(Variation)에 너무 많이 반응하여, 데이터의 실제 의미는 학습하지 못하고 대신 데이터를 "기억하기만 했다"는 것을 의미한다. 수학 교과서를 학교에서 읽기는 했지만, 정작 수학 시험을 볼 때는, 교재에 나와있던 3가지 예에 대한 답만을 알고 있는 것과 같다고 할 수 있다. 선생님이 이런 수학 문제(가령, 2+1=3, 7+2=9, 그리고 4+2=6)를 질문하면, 정답을 맞출 것이다. 그러나 뭔가 다른 것을 질문하면, 예를 들어 1+1=?를 질문

적극적 알고리즘은 명시적 훈련을 사용하지 않는 반면, 소극적 알고리즘은 명시적으로 훈련된다

하면 답을 모른다. 그 이유는 3가지 예에 대한 답은 알고 있지만, 덧셈이 무언지를 학습하지 않았기 때문이다(과거 필자의 담당 교수들에게는 비밀이다, 이 방법이 대학 시절에 필자의 목숨을 살렸다!).

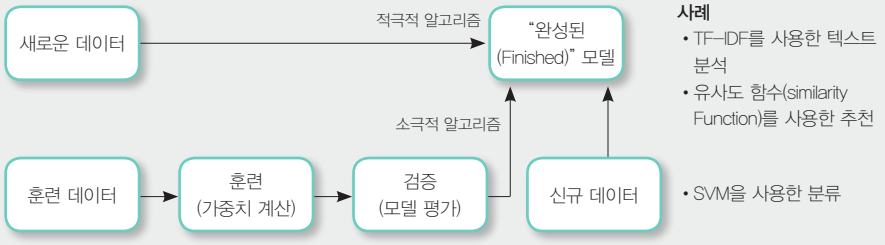
과소적합(고편향)은 뭔가 새로운 것을 학습하기를 거부한다는 점에서 정반대의 문제이다. 10진수로는 덧셈 방법을 알고 있을 수 있다. 그런데 상황이 바뀌어서, 16진수에서 덧셈을 하라는 질문을 받았다. 이때 고 편향을 보인다면, 계속해서 10진수 덧셈을 고수하고 16진수 덧셈을 배우려 하지 않아서, 오답을 얻는다.

표 18 | 모델링: 장애물: 편향과 분산(Variance)



- 저편향과 저분산은 모델이 훈련 데이터로부터 잘 학습했음을 의미하며, 보지 못한 데이터도 일반화 할 수 있음을 의미한다
- 일반적으로, 편향을 줄이면, 분산이 증가하고, 그 반대도 마찬가지다.
- 훈련 표본에서 더 좋은 성과를 내기 위해, 모델 복잡성이 증가하는 경향이 있다.
- 그렇지만, 모델이 과적합을 보이기 시작하면, 훈련 표본의 각 데이터 포인트에 대해 대응하려 한다
- 이는 시험 표본에 대한 모델의 성능을 저하시킨다

표 19 | 훈련을 받거나 받지 않는 머신러닝 예



	적극적 알고리즘	소극적 알고리즘
훈련	명시적 훈련이 전혀 없거나 최소	명시적 훈련
실행 시간	대개 늦음	대개 빠름
메모리 요구사항	더 많은 메모리	더 적은 메모리
이 문서에서 검토한 예	TF–IDF를 사용한 텍스트 분석 유사도 함수(similarity Function)를 사용한 추천	SVM을 사용한 분류

문서와 질의에 대한 TF와 IDF 값이 계산되면, 사용자의 질의와 각 문서의 유사도만 계산하면 된다

두 가지 모두 문제이며, 데이터 과학은 이런 문제를 최소화할 때 도움이 되는 메커니즘을 가지고 있다.

머신러닝 모델 예

두 가지 유형의 알고리즘을 사용하고 있는 머신러닝 예를 두어 가지 살펴보자. 적극적 알고리즘과 소극적 알고리즘. <표 19>는 두 가지 예를 모두 보여주고 있다.

적극적 알고리즘은 명시적 훈련(첫 번째 경로)을 사용하지 않는 반면, 소극적 알고리즘은 명시적으로 훈련된다(표에서 두 번째 경로). 적극적 알고리즘은 명시적으로 훈련되지 않기 때문에, 이 알고리즘의 훈련 단계는 빠르지만(실제로는 아예 존재하지 않는다), 실행은(또는 추론 단계) 훈련된 소극적 알고리즘에 비해 더디다. 적극적 알고리즘은 전체 데이터 세트를 저장해야 하기 때문에 더 많은 메모리를 사용하지만, 소극적 알고리즘을 훈련시킬 때 사용되는 데이터는 훈련이 완료된 다음에는 폐기할 수 있어서, 전체 메모리를 덜 사용한다.

예: TF-IDF를 사용한 문서 검색

텍스트 분석에 적용된 적극적 알고리즘의 첫 번째 예에서, 필자는 TF-IDF라 부르는 알고리즘을 사용한다. 잠시 뒤에 TF와 IDF가 무엇을 의미하는 지 설명하겠지만, 우선 이 예의 목표를 분명히 하자. <표 20>에서 보듯이 5개의 간단하고, 짧은 문서가 있다. 이 문서에 대한 키워드로 이루어진 사전도 있다. 이 사전은 키워드 검색용으로 사용된다. 그리고 질의할 것이 있는 사용자도 있다. 목표는 사용자의 질의에 가장 적합한 문서를 검색하는 것이다. 이 예에서는 관련도가 높은 순서대로 5개의 문서를 제시하고자 한다.

우선, TF와 IDF 약어를 명확하게 설명하자. TF는 Term Frequency의 약자로 어떤 용어(단어)가 얼마나 빈번하게 나타나는지를 나타낸다(즉, 문서에서 해당 단어의 밀집도). TF에 관심을 가지는 이유는 어떤 "중요한" 용어가 더 빈번하게 등장하는 경우, 그 단어가 들어있는 문서가 더 관련 있다고 추정하기 때문이다. TF는 사용자의 질의에 들어있는 단어를 더 관련 있는 문서에 매칭하는데 도움을 준다.

표 20 | 텍스트 분석 예: TF-IDF(Text Frequency-Inverse Document Frequency: 단어 빈도와 역문서 빈도) 문제

항목	내용
문서 1	"그것이 오리처럼 걷고 오리처럼 꽥꽥 운다면, 그것은 오리임에 틀림없다"
문서 2	"북경 오리는 얇고, 바삭거리는 오리 껍질이 가장 소중하게 여겨지며, 이 요리의 정통적인 버전은 대부분 껍데기를 대접한다."
문서 3	"벅스의 스타덤 격상은 워너의 만화영화 제작자들이 벅스가 오리의 질투에 무관심하거나, 그것을 자신의 장점으로 사용하는 동안, 짙투심이 강하고 스포트라이트를 다시 훔쳐오기로 결심한 토끼의 경쟁자를 대피 덕으로 교체하게 만들기도 했다. 이는 이 2인조의 성공 레시피(비결)인 것으로 판명되었다"
문서 4	2007년 7월 1일 오후 6:25 블로그 내용: 나는 cookingforengineers.com에서 와인에 졸인 토끼요리에 대한 레시피를 발견했다."
문서 5	"지난 주 리는 사천식 오리 요리 만드는 법을 보여주었다. 오늘 우리는 내가 지난 여름에 북경에서 맛보았던 인기 있는 요리인 중국식 만두(교자)를 만들어 볼 것이다. 교자 레시피는 여러 가지가 있다."
사전	{북경, 요리, 오리, 토끼, 레시피}
사용자 질의(Query)	"북경 오리 레시피"

문제 정의: 사용자의 질의에 가장 적합한 문서는 어느 것인가, 또는 어떤 순서로 사용자에게 문서를 제시할 것인가?

표 21 | 문서 검색(텍스트 분석): 솔루션

문서

TF 행렬 ⬇

	북경	요리	오리	토끼	레시피
D1	0/3	0/3	3/3	0/3	0/3
D2	1/2	1/2	2/2	0/2	0/2
D3	0/2	0/2	2/2	1/2	1/2
D4	0/1	0/1	0/1	1/1	1/1
D5	1/1	1/1	1/1	0/1	1/1

IDF 벡터 ⬇

	북경	요리	오리	토끼	레시피
D1	0/3=0	0/3=0	3/3=1	0/3=0	0/3=0
D2	1/2=0.5	1/2=0.5	2/2=1	0/2=0	0/2=0
D3	0/2=0	0/2=0	2/2=1	1/2=0.5	1/2=0.5
D4	0/1=0	0/1=0	0/1=0	1/1=1	1/1=1
D5	1/1=1	1/1=1	1/1=1	0/1=0	1/1=1

TF-IDF 행렬

	북경	요리	오리	토끼	레시피
D1	0	0	0.097	0	0
D2	0.199	0.199	0.097	0	0
D3	0	0	0.097	0.199	0.111
D4	0	0	0	0.398	0.222
D5	0.398	0.398	0.097	0	0.222

사용자 질의: "북경 오리 레시피". 질의의 TF-DF 계산

TF ⬇ IDF ⬇

	북경	요리	오리	토끼	레시피
질의	0.398	0	0.097	0	0.222

코사인 유사도 (Cosine Similarity) ⬇

	북경	요리	오리	토끼	레시피	Cos(D, Q)
D1	0	0	0.097	0	0	0.208
D2	0.199	0.199	0.097	0	0	0.639
D3	0	0	0.097	0.199	0.111	0.256
D4	0.398	0.398	0	0.398	0.222	0.232
D5	0.398	0.398	0.097	0	0.222	0.760
질의	.398	0	.097	0	.222	1

⬆

최종 Ordered List

최종 OL: D5, D2, D3, D4, D1

$$sim(\vec{x},\vec{y}) = \frac{\vec{x}\cdot\vec{y}}{|\vec{x}| * |\vec{y}|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

모델 "훈련" 없이, 그저 방정식 몇 개만 적용해도 된다

IDF는 Inverse Document Frequency의 약자이다. 이것은 거의 정반대의 생각이다. 모든 문서에 걸쳐 매우 빈번하게 나타나는 용어는 덜 중요하기 때문에, 이런 용어에 대한 중요도는 줄이고 싶을 것이다. 뻔한 단어는 "a", "an", 그리고 "the"이지만, 특정 주제나 영역에 따라 더 많은 수의 단어들이 있을 것이다. 이런 공통적인 용어를 검색 과정을 혼란스럽게 만드는 잡음으로 간주할 수 있다.

문서와 질의에 대한 TF와 IDF 값이 계산되면, 사용자의 질의와 각 문서의 유사도를 계산하기

만 하면 된다. 유사도 점수(Similarity Score)가 높을수록, 해당 문서는 더 관련이 있다. 그 다음에는 관련 순으로 사용자에게 문서를 제시한다. 쉽지 않은가?

이제 어떻게 하는지를 알았으니, 계산만 하면 된다. <표 21>은 솔루션을 보여준다.

계산 과정을 살펴보자. 이 과정에서는 몇 개의 행렬을 봐야 한다. 머신러닝과 딥 러닝 모델은 행렬 수학을 사용해서 많은 계산을 한다. 데이터 과학자와 함께 일할 때는 이 점에 유의해야 한다. 데이터 과학자는 비즈니스 문제에 적합한 방식으로 이런 유형의 포맷으로 데이터를 얻을 수 있도록 도움을 주어야 할 것이다. 어렵지는 않지만, 이는 데이터 과학 전처리 단계의 기술 중 일부이다.

첫 번째 TF 행렬에서는 각각의 문서에 대해(사전에 지정된 대로) 각 키워드의 정규화("상대") 빈도를 계산한다. 분자(Numerator)는 해당 문서에서 단어 출현 빈도를 나타내고, 분모는 주어진 모든 문서에서 그 단어가 나타난 최대 횟수를 나타낸다. 다른 말로 하면, 모든 분자 전체에서 최대값이다.

두 번째 행렬에서는 사전의 각 용어에 대한 마지막 행에 IDF 벡터(Vector)를 추가한다. 주어진 방정식을 적용하기만 하면 된다.

IDF(t) = log(N/n(t)), 여기에서

- N = 추천할 수 있는 문서 개수
- n(t) = 키워드가 나타난 문서 개수

표 22 | 추천 시스템 예: 유사도 직관(Similarity Intuition)

유형	설명	장점/단점	예: 애완동물
협업적 (Collaborative)	• 다른 사람의 평가에 기반 • 나 같은 사람이 즐기는 아이템을 줘 • 2가지 구현 방식 – 사용자 기반: 예측을 하기 위해 사용자 간의 유사도를 사용 – 아이템 기반: 예측을 하기 위해 아이템 간의 유사도를 사용	• 전문적인 라벨링(Labeling)이 필요 없음 • 다양성이 부족할 수 있음: 기존 사용자의 평가에 의존	• 사용자 기반 – 특정 애완동물 유형을 동반하고 있는 가장 유사한 사람 N명을 찾으라 – 다른 사람의 평가를 예측하기 위해 그들의 평가를 사용하라 • 아이템 기반 – 비슷한 애완동물을 찾아라 – 목표 애완동물에 대한 평가를 예측하기 위해 이런 애완동물에 대한 개인의 평가를 사용하라
콘텐트 기반 (Content–based)	• 단일 사용자와 아이템의 특징 프로파일 기반 • 내가 좋아하는 것과 비슷한 아이템을 줘	• 전문적인 라벨링이 필요하지만 • 어떤 커뮤니티 평가도 필요치 않아서, 커뮤니티가 전혀 필요 없으므로, 콜드 스타트(Cold Start)하기에 좋기 때문에 • 다음 사항에 민감하지 않다. – 특이 취향자(Gray Sheep) (의견이 일관성 있게 다른 그룹과 일치 또는 불일치 하지 않아 유용한 추천을 할 수 없는 사람). – 실링(Shilling) (자신의 제품에 대한 거짓 긍정 평가) – 사용자와 아이템의 증가에 따라 제대로 조절이 되지 않음	• 각 사용자와 개개 애완동물의 유사도를 찾으라 • 유사도에 따라 애완동물의 순서를 정리하라

다음 단계는 문서의 각 열을 마지막 IDF 열로 곱함으로써 문서에 대한 TF-IDF 행렬을 생성하는 것이다. 이제 문서 행렬 작업은 끝났다. 사용자-질의 행렬을 생성하려면 같은 과정을 반복하라.

마지막으로, 2개의 행렬을 결합하고 각 문서와 사용자 질의 간의 유사도를 계산하라. 이 경우, 코사인 유사도라 부르는 유사도를 계산하는 방정식을 사용한다(다른 유사도 계산법을 사용해도 된다). 방정식은 표에 나와있으며, 값은 마지막 열에 있다. 사용자 질의와 자신 간의 유사도 값은 1임에 유의하라. 자신과 비교되기 때문에 1이어야만 한다.

여기서(행렬의 마지막 열에서) 가장 높은 것에서 가장 낮은 순으로 유사도 값을 정렬할 수 있으므로, 이렇게 해서 사용자에게 가장 관련 있는 문서에서 관련이 적은 순서로 문서를 제시한다. 이제 끝났다! 모델의 "훈련"이 없음에 주목하라. 방정식 몇 개만 적용했을 뿐이다.

추천 시스템에서 유사도를 판단하는 2가지 방식: 협업적 접근법, 콘텐트 기반 접근법

예: 협업적 접근방식과 콘텐트 기반 접근방식을 사용한 애완동물 추천

추천 엔진에서 사용되고 있으며, 많은 웹 사이트에서 볼 수도 있는, 적극적 머신러닝 알고리즘의 또 다른 예를 살펴보자. 이 경우, 4명의 애완동물 애호가에 대한 데이터를 가지고 있으며, 그들이 좋아하는 애완동물의 종류에 대한 선호도와 특정 애완동물을 얼마나 좋아하고 있는지를 알고 있다. 선호도에 대해 아는 것이 거의 없는 다섯 번째 애완동물 애호가(에이미)가 존재한다고 가정하자.

목표는 2가지다. 에이미가 특정 애완동물에 줄 것 같은 평가를 예측하고, 에이미의 애완동물 속성에 대한 기호를 알고 있을 경우 에이미가 좋아했을 애완동물의 선호사항을 예측하기. 이 예가 더 잘 알고 있는 사람과 덜 알고 있는 사람 간의 속성을 이용하는, 유사도 문제와 매우 닮아있다는 것을 알 수 있을 것이다.

추천 시스템에서 유사도를 판단하는 방법에는 2가지가 있다. 협업적 방식과 콘텐트 기반 방식이다. 협업 방식은 사용자 기반, 또는 아이템 기반으로 더 세분화할 수 있다.

협업 방식에서는 커뮤니티에 속해 있는 사용자에 대한 평가가 필요하다. 사용자 기반 접근방식에 이런 평가를 적용해서, 커뮤니티에 속해있는 비슷한 사용자의 선호도를 근거로 사용자가 좋아할 것을 예측한다. 반면에, 아이템 기반의 접근방식을 사용해서, 커뮤니티가 좋아하는 아이템 간의 유사도를 근거로 사용자가 좋아하는 예측을 내린다.

콘텐트 기반 방법은 커뮤니티에 속해있는 사용자의 평가를 사용하지 않는다. 대신, 아이템 자체의 특성(Characteristics)을 기반으로 하며, 이런 특성에 할당된 값(라벨, Label)은 특정 분야의 전문가가 제공한다.

각각의 방법은 <표 22>처럼 나름의 장단점을 가지고 있다.

다음 예를 살펴보자. 협업 방식에서는 각 개인의 알려지지 않은 평가를 예측하기 위해 다른 사용자의 애완동물 평가를 사용한다.

우선, 사용자 기반 접근방식을 시도해 보라.

사람의 편견에 의해 왜곡될 수 있는 집계된 개인의 평가를 비교하고 있기 때문에(즉, 개인의

기준치가 각기 다를 수 있기 때문에), 평가를 정규화함으로써(즉, 개개 사용자 평가에서 평가 평균을 뺌으로써) 사용자 편견을 교정하려 하는 피어슨 유사도(Pearson Similarity, 그림의 방정식 참조)라 부르는 유사도 함수를 사용한다. 이 예를 자세히 들여다보면, 앨리스의 평가가 빌의 평가와 가장 비슷하다는 것을 볼 수 있어서, 에이미의 누락된 평가가 빌의 것과 같을 것이라고 가정할 수 있다.

이번에는 아이템 기반 접근방식을 시도해보자. 이 접근방식에서는 개인의 평가에 초점을 두

표 23 | 추천 시스템 예: 유사도 솔루션

협업

	고슴도치	토끼	돼지	개	고양이	사용자 기반 피어슨 유사도
에이미	3	2	5	3	?	
빌	4	3	5	2	5	.72
케이든	2	1	3	4	3	.51
돈	4	5	5	3	5	.21
엠마	3	4	4	1	2	.19
아이템 기반 코사인 유사도[1]	.98	.90	.97	.90		

1 빌, 케이든, 돈, 엠마 만을 근거로 함. 에이미 제외

콘텐트 기반

아이템/사용자	귀여움?	청결함?	껴안아 주고 싶음?	충직함?
고슴도치	4	3	1	1
토끼	5	5	2	4
돼지	1	3	4	2
개	2	3	4	3
고양이	1	2	2	4
에이미	3	3	2	1
빌	1	1	3	4
케이든	2	2	4	3

평가 산출

$r_{u,i} = \frac{1}{N}\sum_{j=1}^{N} r_{j,i}$ (단순 평가)

$r_{u,i} = \overline{r_u} + \alpha \sum_{j=1}^{N} sim(j,u)(r_{j,i} - \overline{r_j})$ (가중치와 중심화된 값)

N: 사용자 u와 가장 가까운 이웃 수
(ru): 아이템 i에 대한 이웃 j의 평가

코사인 유사도를 사용해서 예측된 선호도 순서로 아이템을 추천

	고슴도치	토끼	돼지	개	고양이
에이미	0.96	0.95	0.84	0.88	0.71
빌	0.52	0.74	0.84	0.91	0.96
케이든	0.70	0.83	0.95	0.00	0.91

코사인 유사도

피어슨 유사도

$$sim(\vec{x},\vec{y}) = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}$$

코사인 유사도

$$sim(\vec{x},\vec{y}) = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}} \qquad sim(\vec{x},\vec{y}) = \frac{\vec{x}\cdot\vec{y}}{|\vec{x}|*|\vec{y}|} = \frac{\sum_{i=1}^{n}x_i y_i}{\sqrt{\sum_{i=1}^{n}x_i^2}\sqrt{\sum_{i=1}^{n}y_i^2}}$$

서포트 벡터 머신(SVM: Support Vector Machine)에서는 어떤 아이템이 어느 그룹에 속하는지를 결정할 수 있다

는 대신, 아이템의 평가에 초점을 둔다. 그리고 아이템의 평가는 몇몇 개인이 제공한 복합 평가이기 때문에, 편견에 대해 걱정할 필요가 없어서, 코사인 유사도 함수(표의 방정식 참조)를 사용할 수 있다. 여기서, 고양이가 고슴도치와 가장 유사하다는 것은 알 수 있어서, 고양이에 대한 에이미의 평가가 그녀의 고슴도치에 대한 평가와 비슷할 것이라고 추론할 수 있다.

마지막으로, 콘텐트 기반 접근방식을 시도해보자. 이 접근방식은 커뮤니티 멤버들의 평가를 필요로 하지 않는다. 대신, 전문가가 데이터에 라벨을 붙인다-이 예의 경우, 각 애완동물 유형의 속성(귀여움, 청결함, 껴안아 주고 싶음, 충성심). 각 속성에 대한 개개인의 선호를 알고 있다면, 해당 개인이 가장 기꺼워할 애완동물을 예측하기 위해 코사인 유사도를 사용할 수 있다. 이 예에서 선호도가 낮은 순서로 볼 때, 에이미는 고슴도치, 토끼, 개, 돼지, 그 다음에는 고양이와 즐거운 시간을 보낼 가능성이 높다.

조금 심도 있는 수학으로 들어가 보자. 한가지 예로, 고슴도치에 대한 에이미의 점수를 결정하기 위해, 고슴도치의 애완동물 속성과 에이미가 중요하게 여기는 애완동물 속성간의 유사도를 알아본다.

- 고슴도치의 벡터는 (4,3,1,1)이고
- 에이미의 벡터는 (3,3,2,1)이다
- 이 2개 벡터 간의 유사도를 알아야 한다
- 코사인 유사도 = [4(3) + (3)(3) + (1)(2) + (1)(1)] / [SQRT(4^2 + 3^2 + 1^2 + 1^2) * SQRT(3^2 + 3^2 + 2^2 + 1^2)] = .96

협업 방식의 경우, (평가가 일관되지 않을 수 있는)사용자 전체의 평가를 정규화하기 때문에 피어슨 방정식을 사용한다. 객관적인 평가를 보유하고 있다면, 코사인 유사도를 사용할 수 있다. <표 23>은 솔루션을 보여준다.

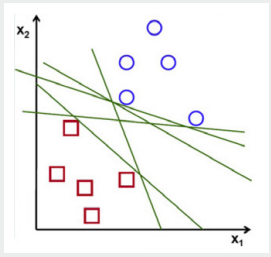
방정식 변수는 다음과 같다.

- u: 사용자
- i: 평가할 아이템
- N: 가장 가까운 이웃 수
- j: 이웃
- rj,I: i에 대한 j의 평가
- rj 평균: j의 평가에 대한 평균
- ru 평균: 사용자의 평가에 대한 평균
- 알파(alpha): 평가에 대한 스케일 팩터(Scaling Factor, 환산 계수). 1은 있는 그대로 사용하라는 의미이다(알파에는 적합한 값이 전혀 없다. 문제 목적과 맥락이 주어졌을 때 숙련된 데이터 과학자가 더 나은 결과를 도출하기 위해 조정할 수 있는 앞서 설명했던 것 같은 하이퍼파라미터 중 한가지다).

표 24 | 분류 예: SVM 문제와 직관

문제
- 몇 가지 속성을 바탕으로 어떤 아이템이 어느 그룹에 속하는 지를 결정

직관
- 새로운 데이터를 어느 세그먼트(클래스)에 둘지를 결정하기 위해 예측 공식을 사용
- 예측 공식은 훈련 데이터를 몇 개의 클래스로 분할하는 커브를 생성함으로써 훈련 데이터로부터 학습한 2개의 변수를 갖는다
  - 정확한 클래스를 결정하기 위한 각 속성(피처)의 중요도(가중치)
  - 지원 벡터(Supported Vector): 클래스를 구분하는 커브에 가장 근접한 훈련 데이터(이 커브를 초평면(Hyperplane)이라 부른다)
- 초평면에 대해 알아보자
  - 초평면의 예: 2개의 피처를 대상으로 하는 초평면(2차원 피처 공간)은 직선이다
  - 초평면의 목표는 모든 세그먼트 간의 거리(차이, Margin)는 최대화하는 동시에 잘못된 클래스에 놓여진(비용 함수로 측정된) 훈련 데이터의 오차를 최소화하는 것이다



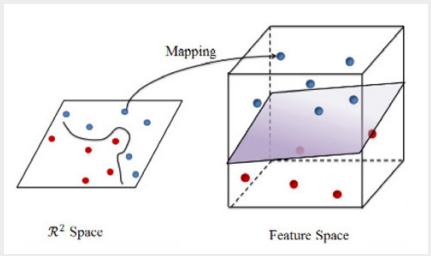
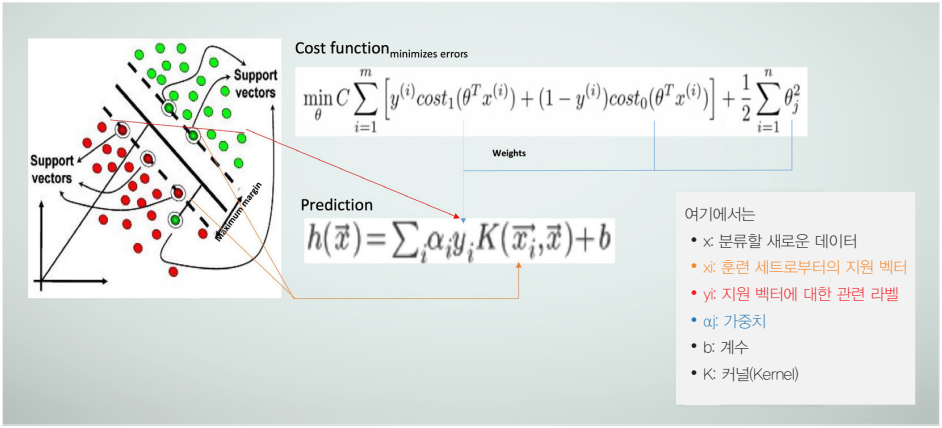


표 25 | 분류 예: SVM 솔루션



예: 서포트 벡터 머신(SVM)을 사용하는 소극적 알고리즘

끝으로, 다음은 서포트 벡터 머신(SBM)이라 부르는 소극적 머신러닝 알고리즘의 예이다. 이 접근방식에서, 사용자는 새로운 고객이 최종적으로는 고수익 고객이나 저수익 고객 중 어느 한 쪽에 속하듯이 어떤 아이템이 어느 그룹에 속하는지를 결정하고 싶어한다. SVM을 사용해서 목적을 달성하려면, 2개의 매개변수를 계산해야 한다.

매개변수를 계산하려면 가능한 데이터 세트를 사용해야 한다. 이것이 데이터 훈련법.

- 각 속성의 가중치(중요도)(속성의 예로는 고객의 수입, 가족 구성원 수, 직업, 그리고 교육 수준이 있다)
- 지원 벡터, 그룹을 구분 짓는(초평면이라 부르는) 커브에 가장 근접해 있는 데이터 세트

그 다음에는 <표 24>처럼 이 2개의 매개변수를 방정식에 대입한다.

견고함, 시장 출시 속도, 그리고 사업 성과물까지 갖춘 AI 솔루션 프레임워크를 구축할 수 있다

이런 매개변수를 계산하는 방법은 사용 가능한 데이터 세트를 사용하는 것이며, 이를 데이터 훈련이라 부른다.

<표 25>는 주어진 예측 라벨 하에서 예측을 위해 사용되는 방정식을 보여준다. 훈련 단계 중에 계산된 값은 다음과 같다.

- 비용 함수를 최소화하기 위해 사용되는 가중치(α와 θ 값).
- 훈련 데이터의 부분 집합인 지원 벡터 xi

모델이 훈련되면, x의 새로운 값(예를 들면, 신규 고객의 속성)을 대입한 다음, x의 새로운 값들이 속하는 클래스 h(x)를 예측할 수 있다(신규 고객이 고 수익 고객으로 예상되는지의 여부).

AI 프로젝트가 실패하는 이유

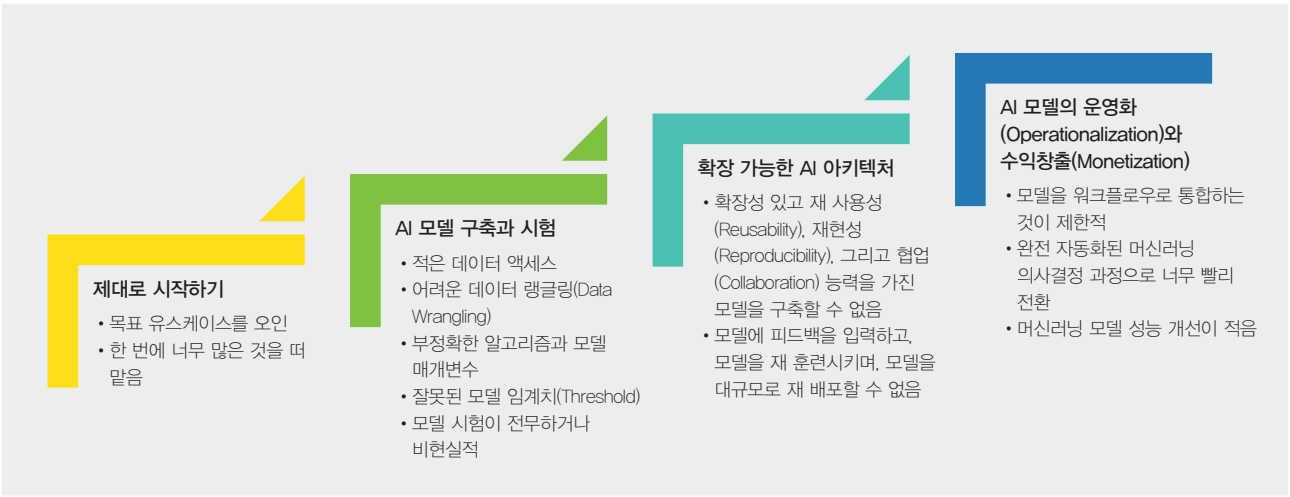
<표 26>과 같이 기업 환경에서 AI 프로젝트가 실패하는 일반적인 이유가 있다. AI 프레임워크라면 항상 이런 함정을 피해야 할 것이다.

첫 번째 실패 요인은 잘못된 유스케이스를 선택했거나, 혹은 충분한 능력이나 인프라 없이 너무 많은 유스케이스를 떠맡은 것이다. AI 솔루션에 더 적합한 문제를 찾아낼 때는 앞에 설명한 범주를 사용할 수 있다. 추가로, 능력과 지식을 점진적으로 쌓아갈 수 있고 기술적 복잡성을 늘려갈 수 있게 해주는 일련의 유스케이스를 설정하는 것이 현명하다.

다음 인력과의 협력으로 올바른 유스케이스를 가장 잘 선택할 수 있다.

- 시험하고 깊어하는 가설뿐 아니라 비즈니스 문제, 맥락, 그리고 제약사항도 알고 있는 사업 부서 스태프
- 비즈니스 목적과 요구사항을 명확하게 해주는 질문을 할 수 있고, 데이터 출처와 데이터 변환을 확인할 수 있는 비즈니스 애널리스트

표 26 | AI 프로젝트가 실패하는 이유



- 머신러닝과 딥 러닝 문제를 간소화하여 모델이 비즈니스의 가설에 대한 답을 제공할 수 있게 해주는 데이터 과학자
- 데이터에 대한 액세스를 제공해줄 수 있는 데이터 엔지니어와 IT 자원

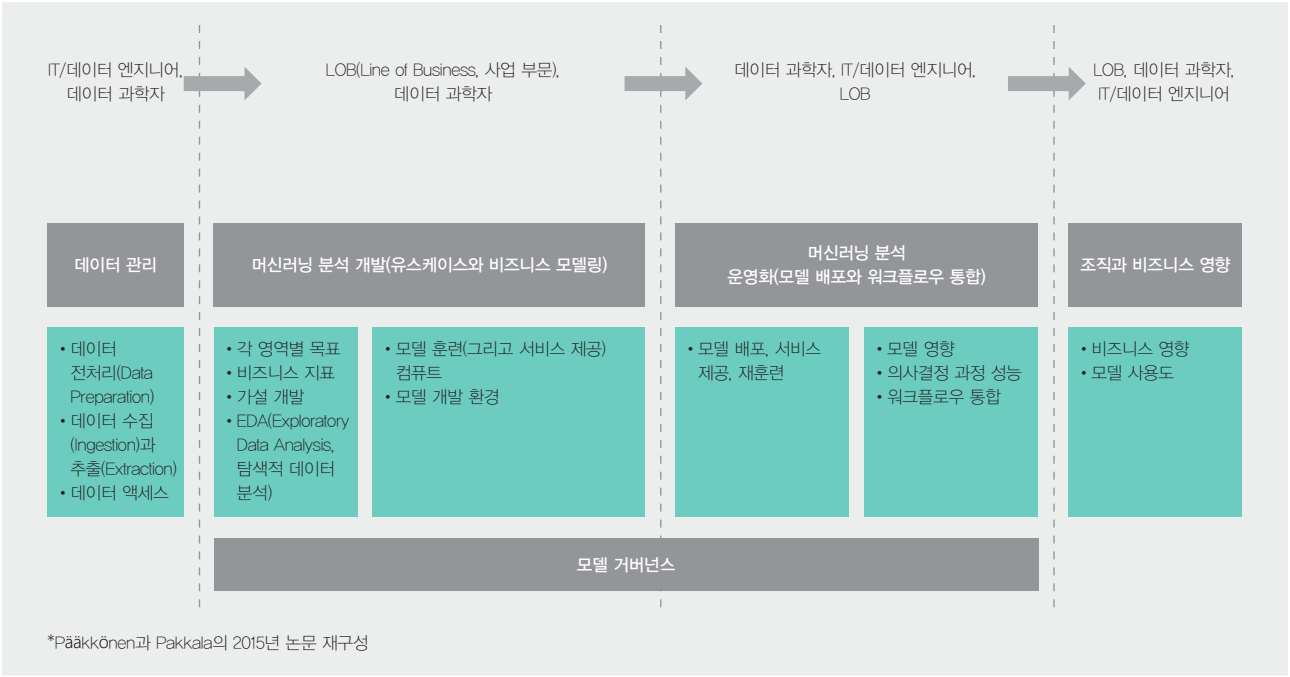
이런 유형의 활동을 사전에 제대로 조직화하고 조율하기 위해서는 다양한 역량을 갖춘 경험 많은 리더가 필요하다. 리더는 비즈니스 영향요소, 운영 상의 추진요소, 워크플로우 장애물과 기회, 데이터 요구사항과 제약, 그리고 활성화 기술을 이해하고 이들간의 균형을 잡을 수 있어야 한다.

두 번째 요인은 AI 모델 자체를 잘못 구축하는 것이다. 이 경우는 2가지로 나뉜다.

- 데이터 과학도 다른 과학과 마찬가지로, 본질적으로 실험적이라고는 하지만(실제로 해보기 전에는 데이터가 실제로 어떤 정보를 알려줄지 전혀 알 수 없다), 데이터 과학에 대한 접근법은 잘 정의되고, 체계가 잡혀 있어야만 하며, 가치 창출 시간을 단축해야 할 것이다.
- 훌륭한 데이터 과학자는 신속하게 시험하고 반복할 수 있으며, 실험을 통해 배울 수 있고, 가능성 있는 것과 비효율적인 접근법을 구분할 수 있으며, 필요한 경우 첨단 기법에 적응할 수 있다. 훌륭한 데이터 과학자는 신속하고, 병행적으로 MVP(Minimal Viable Product, 최소 기능 제품)을 구축한다.

세 번째 요인은 여러 가지 AI 모델을 동시에 신속하게 구축하고 개선하기 위해 확장성 부재이다. 종종, 이 요인은 협력적으로 작업할 수 있고, 데이터 파이프라인(Data Pipeline), 워크플로우, 그리고 모델/알고리즘을 재사용할 수 있으며, 모델 결과를 재현할 능력이 있는 데이터 과학자 문제로 귀결된다. 추가적으로, 더 큰 확장성을 구축하기 위해(시험, 준비 단계, 또는 실무 환경에

표 26 | AI 솔루션 프레임워크



*Pääkkönen과 Pakkala의 2015년 논문 재구성

이제 AI 솔루션 프레임워크 예제로 모든 것을 하나로 묶어보자

서) 운영상의 피드백을 포착해서 신속하게 수용할 수 있어야 한다. 이를 달성하기 위해서는 제대로 된 모델 거버넌스 접근방식뿐 아니라 올바른 인프라 환경도 필요하다.

네 번째 실패 요인은 AI 모델을 운영화하고 수익 창출을 하지 못하는 것이다. 일반적으로 말해서, AI 모델은 다음 두 가지 중 하나를 목적으로 개발된다.

- 이전에 밝혀지지 않은 통찰력을 찾기 위해서
- 의사결정 과정을 자동화하기 위해서(비용 절감과 효율성/생산성을 위해).

의심의 여지없이, 연구소 밖으로 나오지 못한 모델은 이런 작업을 완수할 수 없다.

더 나가서, 모델은 배포될 뿐만 아니라(즉, 사람이나 시스템이 사용할 수 있어야 한다), 운영과 실행 업무에서 실제로 "사용"되도록 워크플로우에 포함되어야만 하며, 예외(모델이 높은 정확도로 의사결정을 하지 못하는 경우) 적절하게 관리돼야만 한다(사람의 개입, 모델 재훈련, 그리고 모델 회수 등). AI 운영을 원활하게 하고 수익을 내려면, 점진적이지만 완벽한 모델 워크플로우 통합, 데이터 입력 값과 모델 성능 매개변수에 대한 감시, 그리고 빈번한 모델 배포 관리를 필요로 한다.

AI의 해답, 엔드투엔드 AI 솔루션 프레임워크

이제 <표 27>처럼, AI 솔루션 예제를 통해 모든 것을 하나로 묶어보자

4가지 구성요소가 있다.

- 데이터 관리.
- 모델 개발.
- 모델 운영화.
- 모델이 사용돼서, 비즈니스에 영향을 미치고, 비즈니스 지표를 개선했는지 확인

첫 번째 구성요소인 데이터 관리는 현 BI 환경의 통상적인 부분이므로, 설명을 하지 않겠다.

두 번째 구성요소인 모델 개발은 크게 2가지 부분으로 이루어져 있다.

- 머신러닝 모델에 적합한 유스케이스를 정의하고 우선순위 부여.
- 확장성 있는 머신러닝 모델 구축.

세 번째 구성요소인 모델 운영화는 모델 배포뿐 아니라 지속적은 재훈련과 재 배포, 운영 워크플로우와 모델의 통합, 그리고 모델 개선을 위한 운영 피드백의 통합까지도 수반한다.

마지막으로, 네 번째 구성요소인 조직과 비즈니스 영향은 간단하고 명확하면서도 조직의 미래 AI 능력 성숙에 필수적이다. 이 구성요소의 기능은 AI 모델이 실제로 LOB에서 사용되고 있으며 비즈니스 결과에 영향을 준다는 점을 확인하는 것이다(즉, LOB가 AI 모델을 신뢰하고 모

표 26 | AI 참조 아키텍처

*Pääkkönen과 Pakkala의 2015년 논문 재구성

견고함, 시장 출시 속도, 그리고 사업 성과물까지 갖춘 AI 솔루션 프레임워크를 구축할 수 있다

델에서 가치를 이끌어내고 있다). LOB의 승인이 없다면, AI에서의 움직임은 거의 일어나지 않을 것이다.

<표 27>의 4가지 구성요소 위에는 협업 그룹이 있다. IT 부서, 데이터 과학자, LOB. AI는 팀 스포츠임을 명심하라.

이 4가지 구성요소는 <표 28>처럼 각각의 요소를 중심으로 하는 참조 아키텍처를 만들 수 있으며, 모델 거버넌스라는 구성요소까지 추가해서 모델 재현성, 데이터 과학 재사용성, 데이터 과학자 협업을 보장하고, 필요한 경우 모델 재 훈련/회수가 가능하도록 설계할 수 있다. 앞서 소개한 아키텍처를 참조해 솔루션을 설계하고 구현하면, 견고함, 시장 출시 속도, 그리고 사업 성과물까지 갖춘 AI 솔루션 프레임워크를 구축할 수 있을 것이다. ITWORLD