



Azure에서 인공 지능 시작

▼ 1100 XP



34분 • 학습 경로 • 모듈 1개

초급 AI 엔지니어 데이터 과학자 개발자 학생 Bot Service Cognitive Services

Machine Learning

AI(인공 지능)는 놀랍고 새로운 솔루션과 환경을 구현합니다. Microsoft Azure를 사용하면 사용하기 쉬운 서비스를 이용해 빠르게 시작할 수 있습니다.

필수 조건

없음

 [책갈피](#) [+ 컬렉션에 추가](#)

이 학습 경로의 모듈



Azure에서 AI 시작

▼ 1100 XP



34분 • 모듈 • 10 단위

4.8 (8,146)

AI를 사용하여 최근까지 공상 과학처럼 보였던 솔루션을 구축함으로써 의료, 재무 관리, 환경 보호 및 기타 영역의 놀라운 발전을 통해 모든 사용자를 위한 더 나은 환경을 구현할 수 있습니다.

개요 ▾



✓ 1100 XP ➔

Azure에서 AI 시작

34분 • 모듈 • 10 단위

★★★★★ 4.8 (8,146) 평가됨 ✓

초급 AI 엔지니어 데이터 과학자 개발자 학생 Bot Service

Cognitive Services Machine Learning

AI를 사용하여 최근까지 공상 과학처럼 보였던 솔루션을 구축함으로써 의료, 재무 관리, 환경 보호 및 기타 영역의 놀라운 발전을 통해 모든 사용자를 위한 더 나은 환경을 구현할 수 있습니다.

학습 목표

이 모듈에서는 AI로 가능해질 수 있는 솔루션 종류와 책임 있는 AI 실행에 대한 고려 사항을 알아봅니다.

📘 책갈피 Ⓛ 컬렉션에 추가

필수 조건

없음

이 모듈은 이러한 학습 경로의 일부입니다.

[Azure에서 인공 지능 시작](#)

AI 소개	✓
-------	---

3분

기계 학습 이해	✓
----------	---

3분

변칙 검색 이해	✓
----------	---

3분

Computer Vision 이해	✓
--------------------	---

3분

자연어 처리 이해	✓
-----------	---

3분

대화형 AI 이해	✓
-----------	---

3분

담당 AI 이해	✓
----------	---

10분

실제로 사용되는 담당 AI 살펴보기



3분

지식 점검



2분

요약



1분

✓ 100 XP



AI 소개

3분

AI를 사용하여 의료 분야를 개선하고, 사용자들이 신체적 결함을 극복하고, 스마트 인프라를 지원하고, 뛰어난 엔터테인먼트 환경을 만들고, 지구 행성을 구할 수 있는 놀라운 소프트웨어를 구축할 수 있습니다.

AI를 사용할 수 있는 몇 가지 방법을 보려면 다음 비디오를 시청하세요.



AI란 무엇인가요?

간단히 말해서, AI는 인간의 행동과 기능을 모방하는 소프트웨어를 만드는 것입니다. 주요 요소는 다음과 같습니다:

- **기계 학습** - 이것은 종종 AI 시스템의 기초이며 예측하고 데이터에서 결론을 도출하기 위해 컴퓨터 모델을 "가르치는" 방법입니다.
- **변칙 검색** - 시스템에서 오류 또는 비정상적인 활동을 자동으로 감지할 수 있는 기능입니다.
- **Computer Vision** - 카메라, 비디오 및 이미지를 통해 세계를 시각적으로 해석할 수 있는 소프트웨어 기능입니다.
- **자연어 처리** - 컴퓨터가 서면 또는 음성 언어를 해석하고 동일하게 응답할 수 있는 기능입니다.
- **대화형 AI** - 대화에 참여하는 소프트웨어 에이전트의 기능입니다.

< 이전

단위 2/10 ▼

다음 >

✓ 100 XP



기계 학습 이해

3분

기계 학습은 AI 솔루션 대부분의 토대입니다.

먼저 기계 학습을 사용하여 어려운 문제를 해결하는 방법에 대한 실제 예제를 살펴보겠습니다.

지속 가능한 경작 기술은 손상된 환경을 보호하는 동시에 식량 생산을 최대화하는 데 필수적입니다. 오스트레일리아에 있는 농업 기술 회사인 *The Yield* 는 센서, 데이터 및 기계 학습을 사용하여 농부들이 날씨, 토양 및 식물 상태와 관련하여 합리적인 의사 결정을 내리도록 도와줍니다.

자세히 알아보려면 다음 비디오를 참조하세요.



[여기](#) 에서는 *The Yield*가 기계 학습을 사용하여 지구를 파괴하지 않으면서 식량을 공급하는 방법을 좀 더 자세히 알아볼 수 있습니다.

기계 학습 작동 방법

그렇다면 기계는 어떻게 학습합니까?

답은 데이터로부터 배운다는 것입니다. 오늘날의 세계에서 우리는 일상 생활을 하면서 엄청난 양의 데이터를 생성합니다. 우리는 문자 메세지, 메일 그리고 소셜 미디어 게시물부터 휴대폰으로 찍은 사진 및 동영상까지 전송합니다. 우리는 엄청난 양의 정보를 생성합니다. 여전히 가정, 자동차, 도시, 대중 교통 인프라 및 공장의 수백만 개의 센서에 의해 많은 데이터가 생성됩니다.

데이터 과학자는 모든 데이터를 사용하여 데이터에서 찾은 관계를 기반으로 예측 및 추론을 할 수 있는 기계 학습 모델을 학습시킬 수 있습니다.

예를 들어 환경 보호 조직이 자원 봉사자가 휴대폰 앱을 사용하여 다양한 종류의 야생화를 식별하고 카탈로그화하기를 원한다고 가정해 보겠습니다. 다음 애니메이션에서는 이러한 시나리오를 활성화하기 위해 어떻게 기계 학습이 사용될 수 있는지를 보여줍니다.

1. 식물학자와 데이터 과학자 팀이 야생화 샘플을 수집합니다.
2. 팀은 샘플을 종별로 정확히 분류하여 레이블을 지정합니다.
3. 레이블이 지정된 데이터는 샘플의 특징과 레이블이 지정된 종 사이의 관계를 찾는 알고리즘을 사용하여 처리됩니다.
4. 알고리즘의 결과는 모델에 요약됩니다.
5. 자원 봉사자에 의해 새로운 샘플이 발견되면 모델은 올바른 종 레이블을 식별할 수 있습니다.

Microsoft Azure의 Machine Learning

Microsoft Azure는 기계 학습 모델을 생성, 관리 및 게시하기 위한 클라우드 기반 플랫폼인 **Azure Machine Learning** 서비스를 제공합니다. Azure Machine Learning은 다음과 같은 특징과 기능들을 제공합니다.

특징	기능
자동화된 기계 학습	이 기능은 비전문가도 데이터를 활용해 효과적인 기계 학습 모델을 빠르게 만들 수 있도록 지원합니다.

특징**기능**

Azure Machine Learning 디자이너

코딩 작업 없이 기계 학습 솔루션을 개발할 수 있는 그래픽 인터페이스

데이터 및 컴퓨팅 관리

전문적인 데이터 과학자들이 데이터 실험 코드를 대규모로 실행하기 위해 활용 할 수 있는 클라우드 기반 데이터 스토리지 및 컴퓨팅 리소스

파이프라인

데이터 과학자, 소프트웨어 엔지니어 및 IT 운영 전문가들이 모델 교육, 배포 및 관리 작업을 구성할 수 있는 파이프라인을 정의할 수 있습니다.

다음 단원: 변칙 검색 이해

계속 >

< 이전

단위 3/10 ▼

다음 >

✓ 100 XP



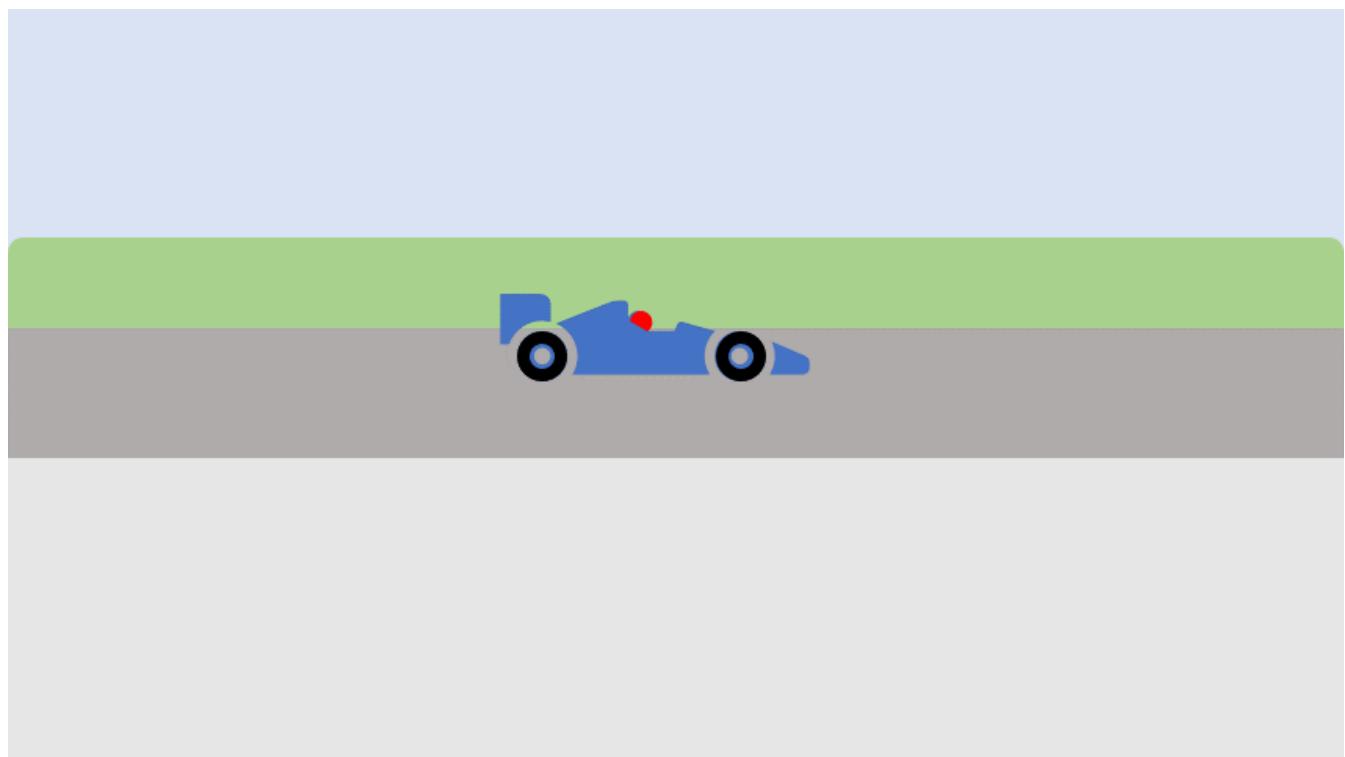
변칙 검색 이해

3분

신용 카드 트랜잭션을 모니터링하고 사기 행위를 나타낼 수 있는 비정상적인 사용 패턴을 감지하는 소프트웨어 시스템을 만드는 경우를 상상해 보세요. 또는 자동화된 프로덕션 라인에서 활동을 추적하고 오류를 식별하는 애플리케이션을 만들 수 있습니다. 또는 센서를 사용하여 잠재적인 기계 고장을 엔지니어에게 사전에 경고하는 레이싱 자동차 원격 분석 시스템을 만들 수도 있습니다.

이러한 종류의 시나리오는 시간에 따라 데이터를 분석하고 비정상적인 변경을 식별하는 기계 학습 기반 기술인 변칙 검색을 사용하여 해결할 수 있습니다.

레이싱 자동차 시나리오에서 변칙 검색이 어떻게 도움이 될 수 있는지 살펴보겠습니다.



1. 차량 센서는 엔진 회전, 브레이크 온도 등의 원격 분석을 수집합니다.
2. 변칙 검색 모델은 시간에 따른 원격 분석 측정의 예상 변동을 이해하도록 학습됩니다.
3. 측정값이 정상적인 예상 범위를 벗어나는 경우 모델은 변칙을 보고하여, 강제로 레이싱이 중단되기 전에 레이싱 엔지니어가 급유 정차 시 운전자를 호출하여 문제를 해결하도록 할 수 있습니다.

Microsoft Azure의 변칙 검색

Microsoft Azure에서 **Anomaly Detector** 서비스는 개발자가 변칙 검색 솔루션을 만드는 데 사용할 수 있는 API(애플리케이션 프로그래밍 인터페이스)를 제공합니다.

자세히 알아보려면 [Anomaly Detector 서비스 웹 사이트](#) 를 참조하세요.

다음 단원: Computer Vision 이해

계속 >

[〈 이전](#)

단위 4/10 ▼

[다음 〉](#)

✓ 100 XP



Computer Vision 이해

3분

Computer Vision은 시각적 처리를 다루는 AI 분야입니다. Computer Vision으로 구현되는 몇 가지 가능성을 살펴보겠습니다.

Seeing AI 앱은 Computer Vision 기능을 잘 보여 주는 예제입니다. 시각 장애인을 위해 디자인된 Seeing AI 앱은 AI의 능력을 이용하여 시각적 세계를 열고 가까이에 있는 사람들, 텍스트 및 물체를 설명합니다.

Seeing AI에 대해 자세히 알아보려면 다음 비디오를 시청하세요.



자세히 알아보려면 [Seeing AI 웹 페이지](#) 를 확인하세요.

Computer Vision 모델 및 기능

대부분의 Computer Vision 솔루션은 카메라, 비디오 또는 이미지의 시각적 입력에 적용할 수 있는 기계 학습 모델을 기반으로 합니다. 다음 표는 Computer Vision의 일반적인 태스크에 대해 설명합니다.

[작업](#)[설명](#)

작업**설명**

이미지
분류

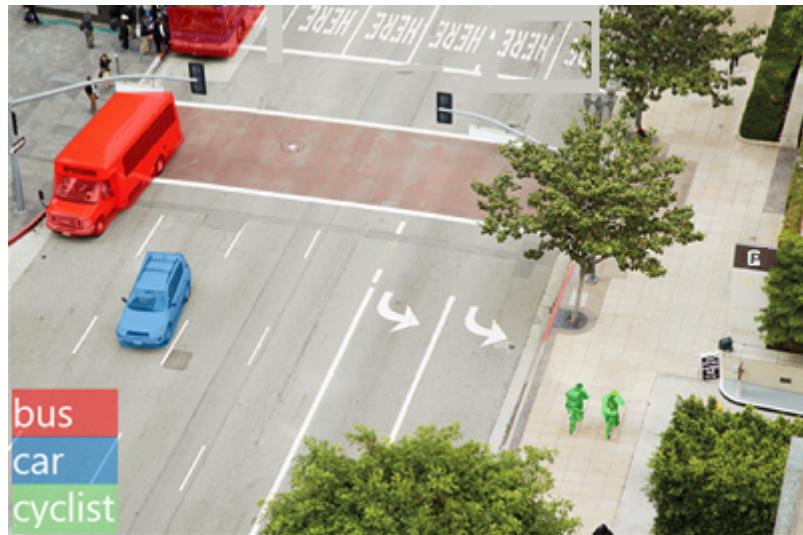


이미지 분류에는 콘텐츠에 따라 이미지를 분류하는 기계 학습 모델 교육이 포함됩니다. 예를 들어 트래픽 모니터링 솔루션에서는 이미지 분류 모델을 사용하여 택시, 버스, 자전거 타는 사람 등 포함된 차량 유형에 따라 이미지를 분류할 수 있습니다.

개체 감지



개체 감지 기계 학습 모델은 이미지 내의 개별 개체를 분류하고 경계 상자를 사용하여 해당 위치를 식별하도록 학습됩니다. 예를 들어 트래픽 모니터링 솔루션은 개체 감지를 사용하여 다양한 차량 클래스의 위치를 식별할 수 있습니다.

작업**설명****의미 체계 구분**

의미 체계 구분은 이미지의 개별 픽셀이 속한 개체에 따라 분류되는 고급 기계 학습 기술입니다. 예를 들어 트래픽 모니터링 솔루션은 "마스크" 레이어로 트래픽 이미지를 오버레이하여 특정 색상으로 다른 차량을 강조 표시할 수 있습니다.

이미지 분석

이미지에 표시된 장면을 요약하는 데 도움이 되는 설명 캡션 또는 이미지를 분류하는 데 도움이 되는 "태그"를 포함하여, 이미지에서 정보를 추출하는 고급 이미지 분석 기술과 기계 학습 모델을 결합한 솔루션을 만들 수 있습니다.

작업 설명

얼굴 감지, 분석 및 인식



얼굴 감지는 이미지에서 사람의 얼굴을 찾는 특수한 형태의 개체 감지입니다. 분류 및 얼굴 기하학 분석 기술과 이를 결합하여 연령 및 감정 상태와 같은 세부 사항을 추론할 수 있으며 얼굴 특징에 따라 개인을 인식할 수도 있습니다.

OCR(광학 문자 인식)



광학 문자 인식은 이미지의 텍스트를 검색하고 읽는 데 사용되는 기술입니다. OCR을 사용하여 사진(예: 도로 표지판 또는 상점 전면)에서 텍스트를 읽거나 문자, 송장 또는 양식과 같은 스캔된 문서에서 정보를 추출할 수 있습니다.

Microsoft Azure의 Computer Vision 서비스

Microsoft Azure는 Computer Vision 솔루션을 만드는 데 도움이 되는 다음과 같은 인지 서비스를 제공합니다.

서비스 기능

Computer Vision	이 서비스를 사용하여 이미지 및 비디오를 분석하고 설명, 태그, 개체 및 텍스트를 추출할 수 있습니다.
-----------------	---

서비스	기능
Custom Vision	이 서비스에서 사용자 고유의 이미지를 사용하여 사용자 지정 이미지 분류 및 개체 검색 모델을 학습시킵니다.
얼굴	Face 서비스를 사용하여 얼굴 감지 및 얼굴 인식 솔루션을 만들 수 있습니다.
Form Recognizer	이 서비스를 사용하여 스캔한 양식 및 청구서에서 정보를 추출할 수 있습니다.

사용 방법

Computer Vision을 사용하여 이미지를 분석하는 방법의 예를 보려면 다음 단계를 수행합니다.

1. 다른 브라우저 탭을 열고 <https://aidemos.microsoft.com/computer-vision>로 이동합니다.
2. 데모 인터페이스를 사용하여 각 단계를 시도합니다. 각 단계에 대해 이미지를 선택하고 Computer Vision 서비스에서 반환하는 정보를 검토할 수 있습니다.

다음 단원: 자연어 처리 이해

계속 >

[〈 이전](#)

단위 5/10 ▼

[다음 〉](#)

✓ 100 XP



자연어 처리 이해

3분

NLP(자연어 처리)는 문어와 구어를 이해하는 소프트웨어 생성을 다루는 AI 영역입니다.

NLP를 사용하면 다음과 같은 소프트웨어를 만들 수 있습니다.

- 문서, 메일 메시지 및 기타 원본의 텍스트를 분석하고 해석합니다.
- 음성 언어를 해석하고 음성 응답을 합성합니다.
- 언어 간 구어 또는 문어 구를 자동으로 번역합니다.
- 명령을 해석하고 적절한 작업을 결정합니다.

예를 들어, *Starship Commander* 는 공상 과학 분야에서 발생하는 Human Interact의 VR(가상 현실) 게임입니다. 이 게임에서는 자연어 처리를 사용하여 플레이어가 내레이션을 제어하고 게임 내 문자 및 우주선 시스템과 상호 작용할 수 있도록 합니다.

자세히 알아보려면 다음 비디오를 시청하세요.



Microsoft Azure의 자연어 처리

Microsoft Azure에서 다음 Cognitive Services를 사용하여 자연어 처리 솔루션을 빌드할 수 있습니다.

[서비스](#)[기능](#)

서비스	기능
Text Analytics	이 서비스를 사용하여 텍스트 문서를 분석하고, 핵심 구를 추출하며, 엔터티(예: 장소, 날짜 및 사람)를 검색하고, 감정(문서가 얼마나 긍정적인지 혹은 부정적인지)을 평가합니다.
Translator Text	이 서비스를 사용하여 텍스트를 60개 이상의 언어로 번역합니다.
Speech	이 서비스를 사용하여 음성을 인식 및 합성하고 음성 언어를 번역합니다.
LUIS(Language Understanding Intelligent Service)	이 서비스를 사용하여 음성 또는 텍스트 기반 명령을 이해할 수 있는 언어 모델을 학습합니다.

사용 방법

자연어를 사용하여 AI 시스템과 상호 작용하는 방법의 예제를 보려면 다음 단계를 수행합니다.

1. 다른 브라우저 탭을 열고 <https://aidemos.microsoft.com/luis/demo>로 이동합니다.
2. 데모 인터페이스를 사용하여 가상 홈의 조명을 제어합니다. 명령을 입력하거나, 마이크 단추를 사용하여 명령을 말하거나, 제안된 문구를 선택하여 시스템이 응답하는 방법을 확인 할 수 있습니다.

다음 단원: 대화형 AI 이해

계속 >

[〈 이전](#)

단위 6/10 ▼

[다음 〉](#)

✓ 100 XP



대화형 AI 이해

3분

대화형 AI는 AI 에이전트가 인간과의 대화에 참여하는 솔루션을 설명하는 데 사용하는 용어입니다. 가장 일반적으로 사용되는 대화형 AI 솔루션은 봇을 사용하여 사용자와의 대화를 관리합니다. 이러한 대화는 웹 사이트 인터페이스, 메일, 소셜 미디어 플랫폼, 메시징 시스템, 전화 통화 및 기타 채널을 통해 진행될 수 있습니다.

봇은 다음에 대한 AI 솔루션의 기반일 수 있습니다.

- 제품 또는 서비스에 대한 고객 지원
- 식당, 항공편, 영화관 및 기타 약속 기반 비즈니스를 위한 예약 시스템
- 의료 상담 및 자체 진단
- 홈 자동화 및 개인용 디지털 비서

조직에서 봇을 사용하여 대화형 AI를 사용하는 방법에 대한 몇 가지 예제는 다음 비디오를 참조하세요.



Microsoft Azure의 대화형 AI

Microsoft Azure에서 대화형 AI 솔루션을 만들려면 다음 서비스를 사용할 수 있습니다.

[서비스](#) [기능](#)

서비스	기능
QnA Maker	이러한 인식 서비스를 사용하여 인간 및 AI 에이전트 간의 대화 기반을 형성할 수 있는 질문 및 답변의 기술 자료를 신속하게 빌드할 수 있습니다.
Azure Bot Service	이 서비스는 봇을 생성, 배포 및 관리하기 위한 플랫폼을 제공합니다. 개발자는 <i>Bot Framework</i> 를 사용하여 봇을 만들고 Azure Bot Service를 사용하여 봇을 관리할 수 있습니다. 또한 QnA Maker 및 LUIS와 같은 백 엔드 서비스를 통합하고 웹 채팅, 메일, Microsoft Teams 등의 채널에 연결할 수 있습니다.

사용 방법

Microsoft 의료 봇은 Azure Bot Service를 기준으로 하며 개발자들이 의료용 대화형 AI 솔루션을 신속하게 만들 수 있도록 합니다. 의료 봇 예제를 보려면 다음을 수행합니다.

1. 다른 브라우저 탭을 열고 <https://www.microsoft.com/research/project/health-bot/>로 이동합니다.
2. 최종 사용자 환경 예제 데모를 사용하는 옵션을 선택합니다.
3. 웹 채팅 인터페이스를 사용하여 봇과 상호 작용합니다.

다음 단원: 담당 AI 이해

계속 >

[〈 이전](#)

단위 7/10 ▼

[다음 〉](#)

✓ 100 XP



담당 AI 이해

10분

인공 지능은 세상에 크게 도움을 주기 위해 활용할 수 있는 강력한 도구입니다. 그러나 다른 도구와 같이 책임감 있게 사용해야 합니다.

Microsoft에서 AI 소프트웨어 개발은 AI 애플리케이션이 의도하지 않은 부정적인 결과 없이 어려운 문제에 대한 놀라운 솔루션을 제공하도록 설계된 6가지 원칙 집합에 의해 안내됩니다.

공정성

AI 시스템은 모든 사람을 공평하게 대해야 합니다. 예를 들어 은행의 대출 승인 애플리케이션을 지원하는 기계 학습 모델을 만든다고 가정합니다. 이 모델은 성별, 민족 또는 특정 지원자 그룹에 대한 부당한 이익 또는 불이익을 초래할 수 있는 다른 요인에 근거한 편견 없이 대출 승인 여부를 예측해야 합니다.

Azure Machine Learning에는 모델을 해석하고 데이터의 각 기능이 모델의 예측에 영향을 미치는 정도를 정량화하는 기능이 포함됩니다. 이 기능을 통해 데이터 과학자와 개발자는 모델의 바이어스를 식별하고 완화할 수 있습니다.

공정성과 관련된 고려 사항에 대한 자세한 내용을 보려면 다음 비디오를 시청하세요.



신뢰성 및 안전성

AI 시스템은 안정적이고 안전한 방식으로 작동해야 합니다. 예를 들어 자율 주행 차량에 대한 AI 기반 소프트웨어 시스템 또는 환자 증상을 진단하고 처방전을 권장하는 기계 학습 모델을 생각해 보세요. 이러한 종류의 시스템을 믿을 수 없다면 인간의 삶에 상당한 위험을 초래할 수 있습니다.

AI 기반 소프트웨어 애플리케이션을 개발할 때는 출시 전에 예상대로 작동하는지 확인하기 위해 엄격한 테스트 및 배포 관리 프로세스를 실시해야 합니다.

신뢰성과 안전성의 고려 사항에 대해 자세 알아 보려면 다음 영상을 시청하세요.



개인 정보 보호 및 보안

AI 시스템은 안전하고 개인 정보를 보호해야 합니다. 기계 학습 모델을 기준으로 하는 AI 시스템은 대량의 데이터에 의존하며, 여기에는 비공개로 유지해야 하는 개인 정보가 포함될 수 있습니다. 모델이 학습되고 시스템이 생산에 들어간 후에도 새 데이터를 사용하여 개인 정보 보호 또는 보안 문제의 적용을 받을 수 있는 예측을 하거나 조치를 취합니다.

개인 정보 및 보안 고려 사항에 대한 자세한 내용은 다음 비디오를 시청하세요.



포용성

AI 시스템은 모든 사람에게 힘을 주고 사람들과 관계를 맺어야 합니다. AI는 신체적 능력, 성별, 성적 지향, 인종 또는 기타 요인에 관계없이 사회의 모든 부분에 혜택을 가져다 주어야 합니다.

포용성과 관련된 고려 사항에 대한 자세한 내용을 보려면 다음 비디오를 시청하세요.



투명성

AI 시스템은 이해하기 쉬워야 합니다. 사용자는 시스템의 목적, 작동 방식 및 예상되는 제한 사항에 대해 충분히 인식해야 합니다.

투명성과 관련된 고려 사항에 대한 자세한 내용을 보려면 다음 비디오를 시청하세요.



책임성

사람들은 AI 시스템에 대한 책임을 져야 합니다. AI 기반 솔루션의 설계자와 개발자는 솔루션이 명확하게 정의된 윤리적 및 법적 표준 충족을 보장하는 거버넌스 및 조직 원칙의 프레임워크 내에서 작동해야 합니다.

책임성과 관련된 고려 사항에 대한 자세한 내용을 보려면 다음 비디오를 시청하세요.



다음 단원: 실제로 사용되는 담당 AI 살펴보기

계속 >

< 이전

단위 8/10 ▼

다음 >

✓ 100 XP



실제로 사용되는 담당 AI 살펴보기

3분

담당 AI의 원칙은 개발자가 윤리적 AI 솔루션을 만들려고 할 때 직면하는 난제를 이해하는 데 도움이 될 수 있습니다. 원칙을 읽어보거나 비디오를 시청하는 것은 쉽지만 원칙을 실행하는 것은 훨씬 더 어려운 일입니다.

사용 방법

다음은 인간과 상호 작용하는 AI 시스템에 원칙이 적용되는 방법을 자세히 알아볼 수 있는 활동입니다.

1. 다른 브라우저 탭을 열고 <https://aidemos.microsoft.com/guidelines-for-human-ai-interaction/demo>로 이동합니다.
2. 인간-AI 상호 작용의 네 단계를 나타내는 카드를 각 데크에서 선택합니다.
 - 초기
 - 상호 작용 중
 - 문제가 발생할 때
 - 시간별로
3. 각 카드에 대해 제공된 예제를 확인합니다.
4. 예제가 반영하는 6가지 원칙을 식별해 보십시오. 하나의 예제가 둘 이상의 원칙을 반영할 수도 있습니다.

추가 리소스

담당 AI 원칙을 구현하는 데 도움이 되는 자세한 내용은

<https://www.microsoft.com/ai/responsible-ai-resources>를 참조하세요.

다음 단원: 지식 점검

[계속 >](#)



Azure Machine Learning을 사용하여 코드 없는 예측 모델 만들기

4300 XP



3시간 29분 • 학습 경로 • 4 모듈

초급 AI 엔지니어 데이터 과학자 Azure Machine Learning

기계 학습은 인공 지능의 핵심이며 많은 최신애플리케이션 및 서비스는 예측 기계 학습 모델에 따라 달라집니다. Azure Machine Learning을 사용하여 코드를 작성하지 않고 모델을 만들고 게시하는 방법을 알아봅니다.

필수 조건

Azure Portal을 탐색할 수 있는 능력

책갈피 컬렉션에 추가

이 학습 경로의 모듈



Azure Machine Learning에서 자동화된 기계 학습 사용

1000 XP



45분 • 모듈 • 9 단위

★★★★★ 4.8 (2,538)

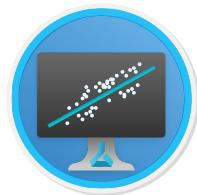
기계 학습 모델을 학습시키는 일은 시간 및 컴퓨팅 리소스가 필요한 반복적인 프로세스입니다. 자동화된 기계 학습을 사용하면 이 작업을 쉽게 수행할 수 있습니다.

개요 ▾

Azure Machine Learning 디자이너를 사용하여

1100 XP





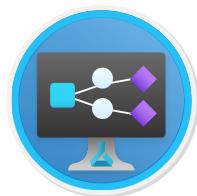
회귀 모델 만들기

55분 • 모듈 • 10 단위

★★★★★ 4.7 (1,645)

회귀는 숫자 값을 예측하는 데 사용되는 지도(supervised) 기계 학습 기법입니다. Azure Machine Learning 디자이너를 사용하여 회귀 모델을 만드는 방법에 대해 알아봅니다.

개요 ▾



Azure Machine Learning 디자이너를 사용하여 분류 모델 만들기

✓ 1100 XP



1 시간 • 모듈 • 10 단위

★★★★★ 4.7 (1,277)

분류는 범주 또는 클래스를 예측하는 데 사용되는 감독형 기계 학습 기술입니다. Azure Machine Learning 디자이너를 사용하여 분류 모델을 만드는 방법에 대해 알아봅니다.

개요 ▾



Azure Machine Learning 디자이너를 사용하여 클러스터링 모델 만들기

✓ 1100 XP



49분 • 모듈 • 10 단위

★★★★★ 4.8 (1,601)

클러스터링은 특징을 기반으로 유사한 엔터티를 그룹화하는 데 사용되는 자율 기계 학습 기술입니다. Azure Machine Learning 디자이너를 사용하여 클러스터링 모델을 만드는 방법을 알아봅니다.

개요 ▾





✓ 1000 XP ➔

Azure Machine Learning에 서 자동화된 기계 학습 사용

45분 • 모듈 • 9 단위

★★★★★ 4.8 (2,538)

평가하기

초급 AI 엔지니어 데이터 과학자 Machine Learning

기계 학습 모델을 학습시키는 일은 시간 및 컴퓨팅 리소스가 필요한 반복적인 프로세스입니다. 자동화된 기계 학습을 사용하면 이 작업을 쉽게 수행할 수 있습니다.

학습 목표

Azure Machine Learning에서 자동화된 기계 학습 사용자 인터페이스를 사용하는 방법 알아보기

📘 책갈피 ⊕ 컬렉션에 추가

필수 조건

Azure Portal 탐색 능력

이 모듈은 이러한 학습 경로의 일부입니다.

[Azure Machine Learning을 사용하여 코드 없는 예측 모델 만들기](#)

소개

1분



기계 학습이란 무엇인가요?

5분



Azure Machine Learning 작업 영역 만들기

5분



컴퓨팅 리소스 만들기

3분



데이터 살펴보기

3분



기계 학습 모델 학습

15분



모델을 서비스로 배포



10분

지식 점검

2분

요약

1분

✓ 100 XP



소개

1분

*Machine Learning*은 대부분의 인공 지능 솔루션의 토대가 되며, 지능형 솔루션을 만드는 과정은 기계 학습을 사용하여 수집한 이전 데이터를 사용하여 예측 모델을 학습시키는 것부터 시작합니다.

'Azure Machine Learning'은 기계 학습 모델을 학습하고 관리하는 데 사용할 수 있는 클라우드 서비스입니다.

이 모듈에서는 다음을 수행하는 방법을 알아봅니다.

- 다양한 종류의 기계 학습 모델을 식별합니다.
- Azure Machine Learning의 자동화된 기계 학습 기능을 사용하여 예측 모델을 학습시키고 배포합니다.

이 모듈을 완료하려면 Microsoft Azure 구독이 필요합니다. 구독이 아직 없다면 <https://azure.microsoft.com>에서 평가판에 가입할 수 있습니다.

다음 단원: 기계 학습이란 무엇인가요?

[계속 >](#)

< 이전

단위 2/9 ▼

다음 >

✓ 100 XP



기계 학습이란 무엇인가요?

5분

Machine Learning은 수학 및 통계를 사용하여 알 수 없는 값을 예측할 수 있는 모델을 만드는 기술입니다.



예를 들어, *Adventure Works Cycles* 는 한 도시에서 자전거를 대여해주는 기업입니다. 이 회사는 충분한 직원과 자전거를 사용할 수 있도록 하기 위해 이전 데이터를 사용하여 일별 대여 수요를 예측하는 모델을 학습시킬 수 있었습니다.

이를 위해 Adventure Works는 특정 일에 대한 정보(요일, 예상 날씨 조건 등)를 입력으로 받은 후 예상되는 대여 수를 출력으로 예측하는 기계 학습 모델을 만들 수 있었습니다.

수학적으로 기계 학습을 항목에 대한 하나 이상의 '기능 (* x')에 연산을 수행하여 예측된 '레이블 (y')을 계산하는 함수(*f)를 정의하는 방법이라고 생각할 수 있습니다. 즉 다음과 같습니다.

$$f(x) = y$$

이 자전거 대여 예제에서는 지정된 날에 대한 세부 정보(요일, 날씨 등)가 '기능***(** x')이고, 해당 날의 대여 수가 '레이블***(** y')이고, 해당 날에 대한 정보를 기준으로 대여 수를 계산하는 함수('f')가 기계 학습 모델에 캡슐화됩니다.

'f' 함수가 'y'를 계산하기 위해 _x*에 대해 수행하는 특정 연산은 만들려는 모델의 유형과 모델을 학습시키는 데 사용되는 특정 알고리즘을 비롯한 다양한 요인에 따라 달라집니다. 또한 대부분의 경우 기계 학습 모델을 학습시키는 데 사용되는 데이터에는 모델 학습을 수행하기 전에 몇 가지 전처리가 필요합니다.

다음 비디오에서는 만들 수 있는 다양한 종류의 기계 학습 모델을 설명하고 모델을 학습시키고 사용하기 위해 일반적으로 수행하는 프로세스를 알아봅니다.



Azure Machine Learning

효과적인 기계 학습 모델을 학습시키고 배포하려면 시간과 리소스가 많이 필요한 많은 작업을 수행됩니다. Azure Machine Learning은 일부 태스크를 간소화하며, 데이터를 준비하고, 모델을 학습시키고, 예측 서비스를 배포하는 데 걸리는 시간을 단축하는 클라우드 기반 서비스입니다. 이 단원의 나머지 부분에서는 Azure Machine Learning과 특히 자동화된 기계 학습 기능에 대해 알아봅니다.

다음 단원: Azure Machine Learning 작업 영역 만들기

계속 >

< 이전

단위 3/9 ▼

다음 >

✓ 100 XP



Azure Machine Learning 작업 영역 만들기

5분

데이터 과학자는 데이터를 탐색 및 전처리하고, 다양한 유형의 모델 학습 알고리즘을 사용하여 정확한 모델을 생성하는 데 많은 노력을 기울입니다. 이러한 작업은 시간이 많이 걸리며 비용이 많이 드는 컴퓨팅 하드웨어가 비효율적으로 사용되는 경우도 많습니다.

Azure Machine Learning은 Azure에서 기계 학습 솔루션을 빌드하고 운영하기 위한 클라우드 기반 플랫폼입니다. 데이터 과학자가 데이터를 준비하고, 모델을 학습하고, 예측 서비스를 게시하고, 사용량을 모니터링하는 데 도움이 되는 여러 특징 및 기능을 포함합니다. 가장 중요한 점은 학습 모델과 관련된 많은 시간이 소요되는 태스크를 자동화하여 데이터 과학자가 효율성을 높이도록 지원하며, 실제로 사용되는 경우에만 비용을 발생시키면서 대량의 데이터를 효과적으로 처리하도록 확장되는 클라우드 기반 컴퓨팅 리소스를 사용할 수 있도록 한다는 것입니다.

Azure Machine Learning 작업 영역 만들기

Azure Machine Learning을 사용하려면 Azure 구독에서 '작업 영역'을 만듭니다. 그런 다음 이 작업 영역을 사용하여 데이터, 컴퓨팅 리소스, 코드, 모델 및 기계 학습 워크로드와 관련된 기타 아티팩트를 관리할 수 있습니다.

① 참고

이 모듈은 **Azure Machine Learning을 사용하여 코드 없는 예측 모델 만들기**의 다른 모듈을 포함하여 Azure Machine Learning 작업 영역을 활용하는 여러 모듈 중 하나입니다. 고유한 Azure 구독을 사용하는 경우 작업 영역을 만든 후 다른 모듈에서 다시 사용하는 것이 좋습니다. 각 모듈을 완료한 후에는 모듈 끝에 있는 **정리** 지침에 따라 컴퓨팅 리소스를 중지해야 합니다.

아직 작업 영역이 없다면 다음 단계에 따라 작업 영역을 만듭니다.

1. Azure 구독과 연결된 Microsoft 자격 증명을 사용하여 [Azure Portal](#)에 로그인합니다.
2. **+ 리소스 만들기**를 선택하고, 'Machine Learning'을 검색하고, 다음 설정을 포함하여 새 Machine Learning 리소스를 만듭니다.

- **구독:** 'Azure 구독'
- **리소스 그룹:** '리소스 그룹 만들기 또는 선택'
- **작업 영역 이름:** '작업 영역의 고유한 이름 입력'
- **지역:** '지리적으로 가장 가까운 지역 선택'

- **스토리지 계정:** '작업 영역에 대해 만들 새로운 기본 스토리지 계정'
- **키 자격 증명 모음:** '작업 영역에 대해 만들 새로운 기본 키 자격 증명 모음'
- **Application insights:** '작업 영역에 대해 만들 새로운 기본 Application Insights 리소스'
- **컨테이너 레지스트리:** 없음('처음으로 컨테이너에 모델을 배포할 때 자동으로 만들 어짐')

3. 작업 영역이 만들어질 때까지 기다립니다(몇 분이 걸릴 수 있음). 그런 다음, 포털로 이동 합니다.
4. 작업 영역의 **개요** 페이지에서 Azure Machine Learning Studio를 실행(또는 새 브라우저 탭을 열고 <https://ml.azure.com> 으로 이동)한 다음, Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다. 메시지가 표시되면 Azure 딜렉터리 및 구독을 선택하고 Azure Machine Learning 작업 영역을 선택합니다.
5. Azure Machine Learning Studio에서 왼쪽 상단에 있는 ⚓ 아이콘을 설정/해제하면 인터페이스에 다양한 페이지가 표시됩니다. 해당 페이지를 사용하여 작업 영역에서 리소스를 관리할 수 있습니다.

Azure Portal을 사용하여 작업 영역을 관리할 수 있지만 데이터 과학자 및 기계 학습 운영 엔지니어의 경우 Azure Machine Learning Studio에서 작업 영역 리소스를 관리하는 데 더욱 초점을 맞춘 사용자 인터페이스를 제공합니다.

다음 단원: 컴퓨팅 리소스 만들기

계속 >

< 이전

단위 4/9 ▼

다음 >

✓ 100 XP



컴퓨팅 리소스 만들기

3분

Azure Machine Learning 작업 영역을 만든 후에는 이 작업 영역을 사용하여 기계 학습 솔루션을 만드는 데 필요한 다양한 자산 및 리소스를 관리할 수 있습니다. 핵심은 Azure Machine Learning이 기계 학습 모델을 학습시키고 관리하기 위한 플랫폼이므로 학습 프로세스를 실행하기 위해 컴퓨팅이 필요하다는 사실입니다.

컴퓨팅 대상 만들기

컴퓨팅 대상은 모델 학습 및 데이터 탐색 프로세스를 실행할 수 있는 클라우드 기반 리소스입니다.

1. [Azure Machine Learning Studio](#)에서 **관리** 아래에 있는 **컴퓨팅** 페이지를 확인합니다. 해당 페이지는 데이터 과학 관련 활동을 위해 컴퓨팅 대상을 관리하는 곳입니다. 다음 네 가지 종류의 컴퓨팅 리소스를 만들 수 있습니다.

- **컴퓨팅 인스턴스**: 데이터 과학자가 데이터 및 모델을 작업하는 데 사용할 수 있는 개발 워크스테이션입니다.
- **컴퓨팅 클러스터**: 실험 코드의 주문형 처리를 지원하는 확장 가능한 가상 머신 클러스터입니다.
- **유추 클러스터**: 학습된 모델을 사용하는 예측 서비스의 배포 대상입니다.
- **연결된 컴퓨팅**: Virtual Machines 또는 Azure Databricks 클러스터와 같은 기존 Azure 컴퓨팅 리소스에 연결합니다.

2. **컴퓨팅 인스턴스** 탭에서 다음 설정을 사용하여 새 컴퓨팅 인스턴스를 추가합니다. 이 인스턴스를 모델을 테스트할 워크스테이션으로 사용합니다.

- **가상 머신 유형**: CPU
- **가상 머신 크기**: Standard_DS11_v2(모든 옵션에서 선택을 선택하여 이 가상 머신 크기를 검색하여 선택함)
- **컴퓨팅 이름**: 고유한 이름 입력
- **SSH 액세스 사용**: 선택 안 함

3. 컴퓨팅 인스턴스가 생성되는 동안 **컴퓨팅 클러스터** 탭으로 전환하고 다음 설정을 사용하여 새 컴퓨팅 클러스터를 추가합니다. 이를 사용하여 기계 학습 모델을 학습합니다.

- **가상 머신 우선 순위**: 전용
- **가상 머신 유형**: CPU

- **가상 머신 크기:** Standard_DS11_v2(모든 옵션에서 선택을 선택하여 이 가상 머신 크기를 검색하여 선택함)
- **컴퓨팅 이름:** 고유한 이름 입력
- **최소 노드 수:** 0
- **최대 노드 수:** 2
- **스케일 다운 전 유휴 시간(초):** 120
- **SSH 액세스 사용:** 선택 안 함

① 참고

이 모듈을 완료하지 않기로 결정한 경우 컴퓨팅 인스턴스를 중지하여 Azure 구독에 불필요한 요금이 발생되지 않도록 해야 합니다.

컴퓨팅 대상은 만들어지는 데 시간이 걸립니다. 대기하는 동안 다음 단원으로 이동할 수 있습니다.

다음 단원: 데이터 살펴보기

계속 >

< 이전

단위 5/9 ▼

다음 >

✓ 100 XP



데이터 살펴보기

3분

기계 학습 모델은 기존 데이터로 학습시켜야 합니다. 이 경우 이전 자전거 대여 세부 정보를 포함하는 데이터 세트를 사용하여 계절 및 기상 특성에 따라 지정된 날짜에 예상되는 자전거 대여 수를 예측하는 모델을 학습시킵니다.

데이터 세트 만들기

Azure Machine Learning에서 모델 학습 및 기타 작업의 데이터는 주로 데이터 세트라는 개체에 캡슐화됩니다.

1. [Azure Machine Learning 스튜디오](#)에서 [데이터 세트 페이지\(자산\)](#)를 확인하고 다음 설정을 사용하여 '웹 파일에서' 새 데이터 세트를 만듭니다.

- **기본 정보***:
 - 웹 URL: <https://aka.ms/bike-rentals>
 - 이름: bike-rentals
 - 데이터 세트 형식: 테이블 형식
 - 설명: 자전거 대여 데이터
- **설정 및 미리 보기**:
 - 파일 형식: 구분 기호로 분리됨
 - 구분 기호: 쉼표
 - 인코딩: UTF-8
 - 열 머리글: 첫 번째 파일의 머리글 사용
 - 행 건너뛰기: 없음
- **스키마**:
 - 경로 이외의 모든 열 포함
 - 자동으로 검색된 형식 검토
- **세부 정보 확인**:
 - 만든 후 데이터 세트를 프로파일링하지 않음

2. 데이터 세트를 만든 후에는 이를 열고 [탐색](#) 페이지를 보면서 데이터 샘플을 확인합니다. 이 데이터에는 자전거 대여에 대한 이전 특징 및 레이블이 포함되어 있습니다.

인용: 이 데이터는 [Capital Bikeshare](#)에서 파생되었으며 게시된 데이터 [사용권 계약](#)에 따라 사용됩니다.

< 이전

단위 6/9 ▼

다음 >

✓ 100 XP



기계 학습 모델 학습

15분

Azure Machine Learning에는 클라우드 컴퓨팅의 확장성을 활용하여 데이터에 가장 적합한 감독형 기계 학습 모델을 찾기 위해 동시에 여러 전처리 기술과 모델 학습 알고리즘을 자동으로 시도하는 자동화된 Machine Learning 기능이 포함되어 있습니다.

① 참고

Azure Machine Learning의 자동화된 Machine Learning 기능은 감독형 기계 학습 모델, 즉 학습 데이터가 알려진 레이블 값을 포함하는 모델을 지원합니다. 자동화된 Machine Learning을 사용하여 다음에 대한 모델을 학습시킬 수 있습니다.

- **분류**(범주 또는 클래스 예측)
- **회귀**(숫자 값 예측)
- **시계열 예측**(시계열 요소를 사용한 회귀를 통해 미래 시점의 숫자 값 예측 가능)

자동화된 Machine Learning 실험 실행

Azure Machine Learning에서 실행하는 작업은 실험이라고 합니다. 자동화된 기계 학습을 사용하여 자전거 대여를 예측하는 회귀 모델을 학습시키는 실험을 실행하려면 다음 단계를 수행합니다.

1. [Azure Machine Learning Studio](#)에서 **자동화된 ML 페이지(작성자)**를 확인합니다.

2. 다음 설정을 사용하여 새 자동화된 ML 실행을 만듭니다.

- **데이터 세트 선택:**
 - **데이터 세트:** bike-rentals
- **실행 구성:**
 - **새 실험 이름:** mslearn-bike-rental
 - **대상 열:** rentals(예측하기 위해 모델을 학습시키는 레이블)
 - **학습 컴퓨팅 대상:** 이전에 만든 컴퓨팅 클러스터
- **태스크 유형 및 설정:**
 - **태스크 유형:** 회귀(모델에서 숫자 값 예측)
 - **추가 구성 설정:**
 - **기본 메트릭:** 정규화된 제곱 평균 오차*(뒷부분에서 이 메트릭에 대해 자세히 소개함)*

- **최적 모델 설명:** 선택함 - 이 옵션을 선택하면 자동화된 기계 학습 기능이 최적 모델의 기능 중요도를 계산합니다. 이를 통해 예측된 레이블에 각 기능이 미치는 영향을 확인할 수 있습니다.
- **차단된 알고리즘:** *RandomForest* 및 *LightGBM* 이외의 모든 알고리즘을 차단합니다. 일반적으로 최대한 많이 시도하려고 하지만 이렇게 하면 시간이 오래 걸릴 수 있습니다.
- **종료 기준:**
 - **학습 작업 시간(시간):** 0.25 - 최대 15분 후에 실험이 종료됩니다.
 - **메트릭 점수 임계값:** 0.08 - 모델이 정규화된 제곱 평균 오차 메트릭 점수 0.08 이하에 도달하는 경우 실험이 종료됩니다.
- **기능화 설정:**
 - **기능화 사용:** 선택함 - *Azure Machine Learning* 기능은 학습 전에 자동으로 기능을 전처리합니다.

3. 자동화된 ML 실행 세부 정보의 제출을 완료하면 자동으로 시작됩니다. 실행 상태가 '준비 중'에서 '실행 중'으로 변경될 때까지 기다립니다.
4. 실행 상태가 '실행 중'으로 변경되면 **모델** 탭에서 학습 알고리즘과 전처리 단계의 가능한 각 조합이 시도되고 결과 모델의 성능이 평가되는지 관측합니다. 이 페이지는 주기적으로 자동으로 새로 고쳐지지만 ↴ **새로 고침** 을 선택할 수도 있습니다. 클러스터 노드를 초기화해야 학습을 시작할 수 있으므로 모델이 표시되는 데 10분 정도 걸릴 수 있습니다.
5. 실험이 완료될 때까지 기다립니다. 시간이 걸릴 수 있으므로 지금 잠깐 쉬어가는 것이 좋을 수 있습니다.

최적 모델 검토

실험을 완료한 후 생성된 최적 모델을 검토할 수 있습니다. 이 경우에는 종료 조건을 사용하여 실험을 중지했습니다. 따라서 실험에서 발견한 "최적" 모델이 가장 적합한 모델이 아니고 이 연습에 허용되는 시간 안에 확인된 최적 모델에 불과할 수 있습니다.

1. 자동화된 기계 학습 실행의 **세부 정보** 탭에서 최적 모델 요약을 확인합니다.
2. 최적 모델의 **알고리즘 이름** 을 선택하여 세부 정보를 확인합니다.

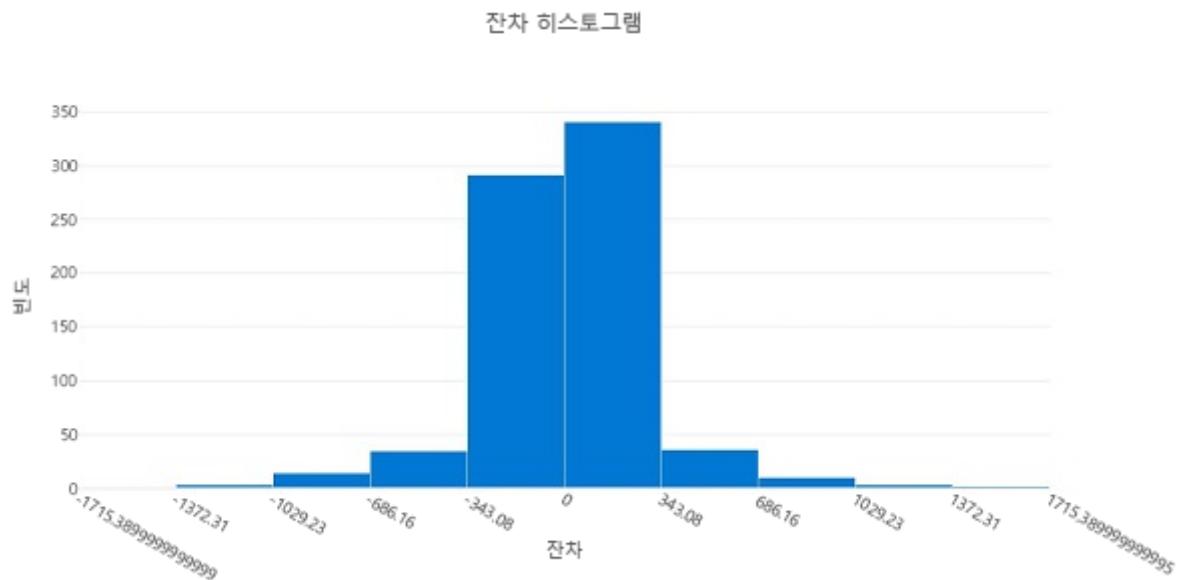
최적 모델은 지정한 평가 메트릭에 따라 식별됩니다(정규화된 제곱 평균 오차). 이 메트릭을 계산하기 위해 학습 프로세스에서는 일부 데이터를 사용하여 모델을 학습시키고, 교차 유효성 검사라는 기술을 적용하여 학습시킨 모델을 학습되지 않은 데이터로 반복적으로 테스트하고 예측 값을 실제 알려진 값과 비교했습니다. 예측 값과 실제 값 간 차이(잔차)는 모델의 오차 크기를 나타내며, 이 특정 성능 메트릭은 모든 테스트 사례에서 오차를 제곱하고, 이러한 제곱의 평균을 구한 후 제곱근을 구하는 방식으로 계산됩니다. 이 메트릭은 이 값이 작을수록 모델이 보다 정확하게 예측한다는 것을 의미합니다.

3. 정규화된 제곱 평균 오차 값 옆에 있는 **다른 모든 메트릭 보기**를 선택하여 회귀 모델에 대한 가능한 다른 평가 메트릭 값을 확인합니다.
4. **메트릭** 탭을 선택하고 **잔차** 및 **predicted_true** 차트를 아직 선택하지 않은 경우 선택합니다. 그런 다음, 예측 값을 참 값과 비교하고 잔차(예측 값과 실제 값의 차이)를 히스토그램으로 표시하여 모델의 성능을 보여 주는 차트를 검토합니다.

예측 값 및 참 값 차트는 예측 값이 참 값과 밀접하게 상호 연관되는 대각선 추세를 표시합니다. 점선은 완벽한 모델의 작동 성능을 보여 주며, 모델의 평균 예측 값 줄이 이 점선에 가까울수록 성능이 좋은 것입니다. 꺾은선형 차트 아래의 히스토그램은 참 값의 분포를 보여 줍니다.

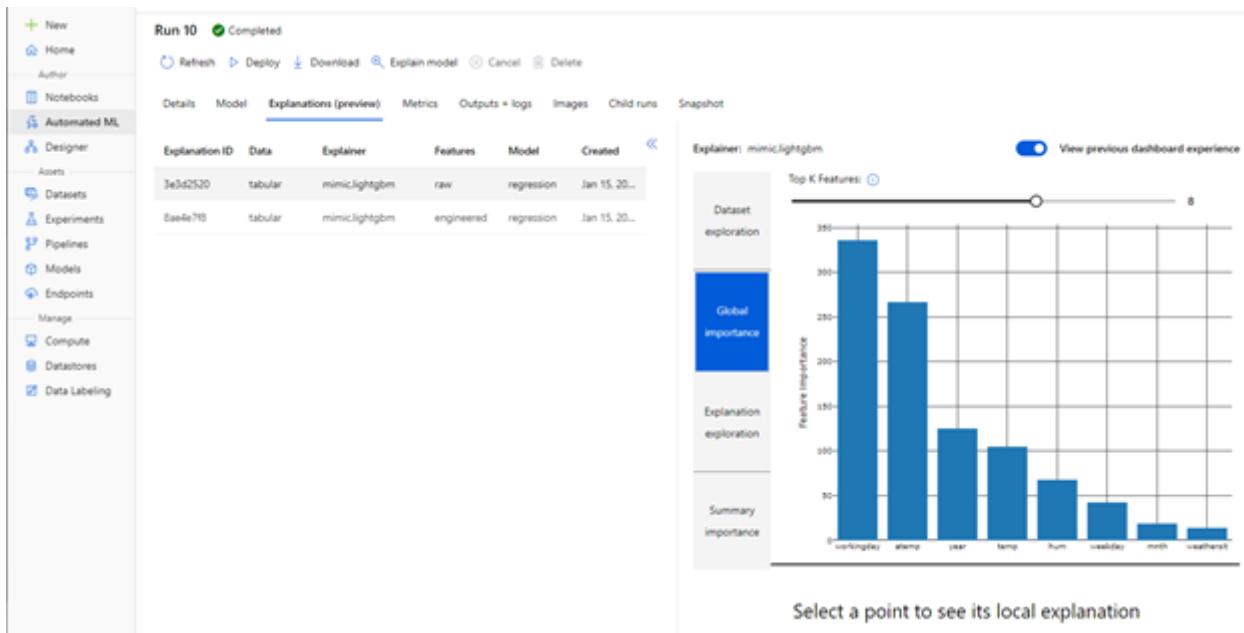


잔차 히스토그램은 잔차 값 범위의 빈도를 보여 줍니다. 잔차는 모델로 설명할 수 없는 예측 값과 실제 값 간의 차이를 나타냅니다. 즉, 오차를 나타냅니다. 따라서 가장 자주 발생하는 잔차 값은 0 주변에 클러스터링되고(즉, 대부분의 오차는 작음), 오차가 아주 클 가능성은 훨씬 작습니다.



5. **설명** 탭을 선택합니다. 설명 ID 옆의 화살표 >>를 클릭하여 설명 목록을 펼칩니다. 설명 ID를 선택하고 오른쪽에서 **이전 대시보드 환경 보기**를 선택합니다. 그런 다음 글로벌 중

요도를 선택합니다. 이 차트는 다음과 같이 데이터 세트의 각 기능이 레이블 예측에 영향을 미치는 정도를 보여 줍니다.



다음 단원: 모델을 서비스로 배포

계속 >

< 이전

단위 7/9 ▼

다음 >

✓ 100 XP



모델을 서비스로 배포

10분

자동화된 기계 학습을 사용하여 일부 모델을 학습시킨 후에는 클라이언트 애플리케이션에서 사용할 수 있는 최상의 모델을 서비스로 배포할 수 있습니다.

예측 서비스 배포

Azure Machine Learning에서 서비스를 ACI(Azure Container Instances) 또는 AKS(Azure Kubernetes Service) 클러스터로 배포할 수 있습니다. 프로덕션 시나리오의 경우 AKS를 배포하는 것이 좋습니다. 이 경우 유추 클러스터 컴퓨팅 대상을 만들어야 합니다. 이 연습에서는 테스트에 적합한 배포 대상인 ACI 서비스를 사용하며, 유추 클러스터를 만들 필요가 없습니다.

1. [Azure Machine Learning Studio](#) 의 **자동화된 ML** 페이지에서 자동화된 기계 학습 실험에 대해 실행을 선택하고 **세부 정보** 탭을 봅니다.
2. 최적 모델의 알고리즘 이름을 선택합니다. 그런 다음, **모델** 탭에서 **배포** 단추를 사용하여 다음 설정으로 모델을 배포합니다.
 - **이름:** predict-rentals
 - **설명:** 자전거 대여 예측
 - **컴퓨팅 형식:** ACI
 - **인증 사용:** 선택됨
3. 배포가 시작될 때까지 대기합니다. 이 작업에는 몇 초 정도 걸릴 수 있습니다. 그런 다음, **모델 요약** 섹션에서 **predict-rentals** 서비스에 대한 **배포 상태**를 확인합니다. 이 상태는 **실행 중** 이어야 합니다. 이 상태가 **성공**으로 변경될 때까지 기다립니다. 주기적으로 **새로 고침**을 선택해야 할 수도 있습니다.
4. Azure Machine Learning Studio에서 [엔드포인트](#) 페이지를 보고 **predict-rentals** 실시간 엔드포인트를 선택합니다. 그런 다음, **사용** 탭을 선택하고 다음 정보를 확인합니다. 클라이언트 애플리케이션에서 배포된 서비스를 연결하려면 이 정보가 필요합니다.
 - 서비스에 대한 REST 엔드포인트
 - 서비스에 대한 기본 키
5. 이 값 옆에 있는 링크를 사용하여 이를 클립보드에 복사할 수 있습니다.

배포된 서비스 테스트

이제 서비스를 배포했으므로 간단한 코드를 사용하여 테스트할 수 있습니다.

1. 브라우저에서 **predict-rentals** 서비스 페이지의 **사용** 페이지가 열린 상태에서 새 브라우저 탭을 열고 **Azure Machine Learning Studio** 의 두 번째 인스턴스를 엽니다. 그런 다음, 새 탭에서 **Notebooks** 페이지(작성자 아래)를 봅니다.
2. **Notebooks** 페이지의 **내 파일**에서 □ 단추를 사용하여 다음 설정으로 새 파일을 만듭니다.
 - **파일 위치:** Users/사용자 이름
 - **파일 이름:** Test-Bikes
 - **파일 형식:** Notebook
 - **이미 있는 경우 덮어쓰기:** 선택됨
3. 새 Notebook이 만들어지면 이전에 만든 컴퓨팅 인스턴스가 **컴퓨팅** 상자에서 선택되었는지, 그리고 상태가 **실행 중**인지 확인합니다.
4. << 단추를 사용하여 파일 탐색기 창을 축소하고 **Test-Bikes.ipynb** Notebook 탭에 사용할 수 있는 공간을 확보합니다.
5. Notebook에서 만들어진 사각형 셀에 다음 코드를 붙여넣습니다.

Python

복사

```

endpoint = 'YOUR_ENDPOINT' #Replace with your endpoint
key = 'YOUR_KEY' #Replace with your key

import json
import requests

#An array of features based on five-day weather forecast
x = [[1,1,2022,1,0,6,0,2,0.344167,0.363625,0.805833,0.160446],
      [2,1,2022,1,0,0,0,2,0.363478,0.353739,0.696087,0.248539],
      [3,1,2022,1,0,1,1,1,0.196364,0.189405,0.437273,0.248309],
      [4,1,2022,1,0,2,1,1,0.2,0.212122,0.590435,0.160296],
      [5,1,2022,1,0,3,1,1,0.226957,0.22927,0.436957,0.1869]]

#Convert the array to JSON format
input_json = json.dumps({"data": x})

#Set the content type and authentication for the request
headers = {"Content-Type": "application/json",
           "Authorization": "Bearer " + key}

#Send the request
response = requests.post(endpoint, input_json, headers=headers)

#If we got a valid response, display the predictions
if response.status_code == 200:
    y = json.loads(response.json())
    print("Predictions:")
    for i in range(len(x)):
        print (" Day: {}. Predicted rentals: {}".format(i+1, max(0,
round(y["result"][i]))))

```

```
else:  
    print(response)
```

① 참고

코드의 세부 정보에 대해 너무 걱정하지 마세요. 가상 날씨 예측 데이터를 사용하여 5 일 간의 특징을 정의하고, 만든 **predict-rentals** 서비스를 사용하여 5일 간의 자전거 대여 수를 예측합니다.

6. **predict-rentals** 서비스의 **사용** 페이지가 포함된 브라우저 탭으로 전환한 다음 서비스에 대한 REST 엔드포인트를 복사합니다. Notebook을 포함하는 탭으로 돌아가 키를 코드에 복사하고 YOUR_ENDPOINT를 대체합니다.
7. **predict-rentals** 서비스의 **사용** 페이지가 포함된 브라우저 탭으로 전환한 다음 서비스에 대한 기본 키를 복사합니다. Notebook을 포함하는 탭으로 돌아가 키를 코드에 복사하고 YOUR_KEY를 대체합니다.
8. Notebook을 저장한 다음 셀 옆에 있는 ▶ 단추를 사용하여 코드를 실행합니다.
9. 5일 기간에 포함된 각 날의 예상 대여 수가 반환되는지 확인합니다.

다음 단원: 지식 점검

계속 >



✓ 1100 XP ➔

Azure Machine Learning 디자이너를 사용하여 회귀 모델 만들기

55분 • 모듈 • 10 단위

★★★★★ 4.7 (1,645)

평가하기

초급 AI 엔지니어 데이터 과학자 Machine Learning

회귀는 숫자 값을 예측하는 데 사용되는 지도(supervised) 기계 학습 기법입니다. Azure Machine Learning 디자이너를 사용하여 회귀 모델을 만드는 방법에 대해 알아봅니다.

학습 목표

Azure Machine Learning 디자이너를 사용하여 회귀 모델을 학습하고 게시하는 방법에 대해 알아봅니다.

📘 책갈피 Ⓛ 컬렉션에 추가

필수 조건

Azure Portal 탐색 능력

이 모듈은 이러한 학습 경로의 일부입니다.

[Azure Machine Learning을 사용하여 코드 없는 예측 모델 만들기](#)

소개

2분



Azure Machine Learning 작업 영역 만들기

5분



컴퓨팅 리소스 만들기

5분



데이터 살펴보기

10분



학습 파이프라인 만들기 및 실행

12분



회귀 모델 평가

5분



유추 파이프라인 만들기

8분



예측 서비스 배포

5분



지식 점검

2분



요약

1분





소개

2분

'회귀'란 항목의 '특징'을 기반으로 숫자 '레이블'을 예측하는 데 사용되는 기계 학습의 한 가지 형태입니다. 예를 들어 자동차 판매 회사는 자동차의 특성(예: 엔진 크기, 좌석 수, 주행 거리)을 사용하여 가능한 판매 가격을 예측할 수 있습니다. 이 경우 자동차의 특성이 특징이며 판매 가격은 레이블입니다.



회귀는 모델이 학습을 통해 특징 조합을 레이블에 '맞추도록' 레이블에 대한 특징과 알려진 값을 모두 포함하는 데이터를 사용하여 모델을 학습하는 '지도(supervised)' 기계 학습 기법의 예입니다. 그리고 나서, 학습이 완료된 후 학습된 모델을 사용하여 레이블이 알려지지 않은 새 항목에 대한 레이블을 예측할 수 있습니다.

Microsoft Azure Machine Learning 디자이너를 사용하여 코드를 작성할 필요 없이 끌어다 놓기 시각적 인터페이스를 사용하여 회귀 모델을 만들 수 있습니다.

이 모듈에서는 다음 방법을 알아봅니다.

- Azure Machine Learning 디자이너를 사용하여 회귀 모델을 학습합니다.
- 유추에 회귀 모델을 사용합니다.
- 회귀 모델을 서비스로 배포합니다.

이 모듈을 완료하려면 Microsoft Azure 구독이 필요합니다. 구독이 아직 없다면 <https://azure.microsoft.com>에서 평가판에 가입할 수 있습니다.

다음 단원: Azure Machine Learning 작업 영역 만들기

계속 >

< 이전

단위 2/10 ▼

다음 >

✓ 100 XP



Azure Machine Learning 작업 영역 만들기

5분

Azure Machine Learning은 Azure에서 기계 학습 솔루션을 빌드하고 운영하기 위한 클라우드 기반 플랫폼입니다. 데이터 과학자가 데이터를 준비하고, 모델을 학습하고, 예측 서비스를 게시하고, 사용량을 모니터링하는 데 도움이 되는 여러 특징 및 기능을 포함합니다. 해당 특징 중 하나는 '디자이너'라는 시각적 인터페이스로, 코드 작성 없이 기계 학습 모델을 학습, 테스트 및 배포할 수 있게 해 줍니다.

Azure Machine Learning 작업 영역 만들기

Azure Machine Learning을 사용하려면 Azure 구독에서 '작업 영역'을 만듭니다. 그런 다음 이 작업 영역을 사용하여 데이터, 컴퓨팅 리소스, 코드, 모델 및 기계 학습 워크로드와 관련된 기타 아티팩트를 관리할 수 있습니다.

ⓘ 참고

이 모듈은 Azure Machine Learning 작업 영역을 사용하는 여러 가지 모듈 중 하나입니다.

Azure AI 기본 사항 또는 **Azure 데이터 과학자** 인증을 준비하는 과정에서 이 모듈을 진행하는 경우, 작업 영역을 한 번 만들어 두고 다른 모듈에서 재사용하는 것이 좋습니다. 각 모듈을 완료한 후에는 모듈 끝에 있는 정리 지침에 따라 컴퓨팅 리소스를 중지해야 합니다.

아직 없는 경우 다음 단계에 따라 작업 영역을 만듭니다.

1. Microsoft 자격 증명을 사용하여 [Azure Portal](#) 에 로그인합니다.
2. + 리소스 만들기를 선택하고, 'Machine Learning'을 검색하고, 다음 설정을 포함하여 새 Machine Learning 리소스를 만듭니다.

- **구독:** 'Azure 구독'
- **리소스 그룹:** '리소스 그룹 만들기 또는 선택'
- **작업 영역 이름:** '작업 영역의 고유한 이름 입력'
- **지역:** '지리적으로 가장 가까운 지역 선택'
- **스토리지 계정:** '작업 영역에 대해 만들 새로운 기본 스토리지 계정'
- **키 자격 증명 모음:** '작업 영역에 대해 만들 새로운 기본 키 자격 증명 모음'
- **Application insights:** '작업 영역에 대해 만들 새로운 기본 Application Insights 리소스'
- **컨테이너 레지스트리:** 없음('처음으로 컨테이너에 모델을 배포할 때 자동으로 만들 어짐')

3. 작업 영역이 만들어질 때까지 기다립니다(몇 분이 걸릴 수 있음). 그런 다음 포털에서 해당 위치로 이동합니다.
4. 작업 영역의 **개요** 페이지에서 Azure Machine Learning Studio를 실행(또는 새 브라우저 탭을 열고 <https://ml.azure.com> 으로 이동)한 다음 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다.
5. Azure Machine Learning Studio에서 왼쪽 상단에 있는 ☰ 아이콘을 설정/해제하면 인터페이스에 다양한 페이지가 표시됩니다. 해당 페이지를 사용하여 작업 영역에서 리소스를 관리할 수 있습니다.

Azure Portal을 사용하여 작업 영역을 관리할 수 있지만 데이터 과학자 및 기계 학습 운영 엔지니어의 경우 Azure Machine Learning Studio에서 작업 영역 리소스를 관리하는 데 특화된 사용자 인터페이스를 이용할 수 있습니다.

다음 단원: 컴퓨팅 리소스 만들기

계속 >

< 이전

단위 3/10 ▼

다음 >

✓ 100 XP



컴퓨팅 리소스 만들기

5분

Azure Machine Learning 디자이너를 사용하여 모델을 학습하고 배포하려면 학습 프로세스를 실행하고, 배포 후 학습된 모델을 테스트하기 위한 컴퓨팅이 필요합니다.

컴퓨팅 대상 만들기

컴퓨팅 대상은 모델 학습 및 데이터 탐색 프로세스를 실행할 수 있는 클라우드 기반 리소스입니다.

1. [Azure Machine Learning Studio](#)에서 관리 아래에 있는 컴퓨팅 페이지를 확인합니다. 해당 페이지는 데이터 과학 관련 활동을 위해 컴퓨팅 대상을 관리하는 곳입니다. 다음 네 가지 종류의 컴퓨팅 리소스를 만들 수 있습니다.

- **컴퓨팅 인스턴스**: 데이터 과학자가 데이터 및 모델을 작업하는 데 사용할 수 있는 개발 워크스테이션입니다.
- **컴퓨팅 클러스터**: 실험 코드의 주문형 처리를 지원하는 확장 가능한 가상 클러스터입니다.
- **유추 클러스터**: 학습된 모델을 사용하는 예측 서비스의 배포 대상입니다.
- **연결된 컴퓨팅**: Virtual Machines 또는 Azure Databricks 클러스터와 같은 기존 Azure 컴퓨팅 리소스에 연결합니다.

2. 컴퓨팅 인스턴스 탭에서 다음 설정을 사용하여 새 컴퓨팅 인스턴스를 추가합니다. 이 인스턴스를 모델을 테스트할 워크스테이션으로 사용합니다.

- **가상 머신 유형**: CPU
- **가상 머신 크기**: Standard_DS11_v2(모든 옵션에서 선택을 선택하여 이 가상 머신 크기를 검색하여 선택함)
- **컴퓨팅 이름**: 고유한 이름 입력
- **SSH 액세스 사용**: 선택 안 함

3. 컴퓨팅 인스턴스가 생성되는 동안 컴퓨팅 클러스터 탭으로 전환하고 다음 설정을 사용하여 새 컴퓨팅 클러스터를 추가합니다. 이를 사용하여 기계 학습 모델을 학습합니다.

- **가상 머신 우선 순위**: 전용
- **가상 머신 유형**: CPU
- **가상 머신 크기**: Standard_DS11_v2(모든 옵션에서 선택을 선택하여 이 가상 머신 크기를 검색하여 선택함)

- **컴퓨팅 이름:** 고유한 이름 입력
- **최소 노드 수:** 0
- **최대 노드 수:** 2
- **스케일 다운 전 유휴 시간(초):** 120
- **SSH 액세스 사용:** 선택 안 함

① 참고

이 모듈을 완료하지 않기로 결정한 경우 컴퓨팅 인스턴스를 중지하여 Azure 구독에 불필요한 요금이 발생되지 않도록 해야 합니다.

컴퓨팅 대상은 만들어지는 데 시간이 걸립니다. 대기하는 동안 다음 단원으로 이동할 수 있습니다.

다음 단원: 데이터 살펴보기

계속 >

< 이전

단위 4/10 ▼

다음 >

✓ 100 XP



데이터 살펴보기

10분

회귀 모델을 학습하려면, 기존 '특징'(예측의 대상이 되는 엔터티의 특성)과 알려진 '레이블' 값(모델을 학습하여 예측할 숫자 값)이 포함된 데이터 세트가 필요합니다.

파이프라인 만들기

Azure Machine Learning 디자이너를 사용하려면 기계 학습 모델을 학습하는 데 사용할 파이프라인을 만듭니다. 해당 파이프라인은 모델을 학습할 데이터 세트로 시작합니다.

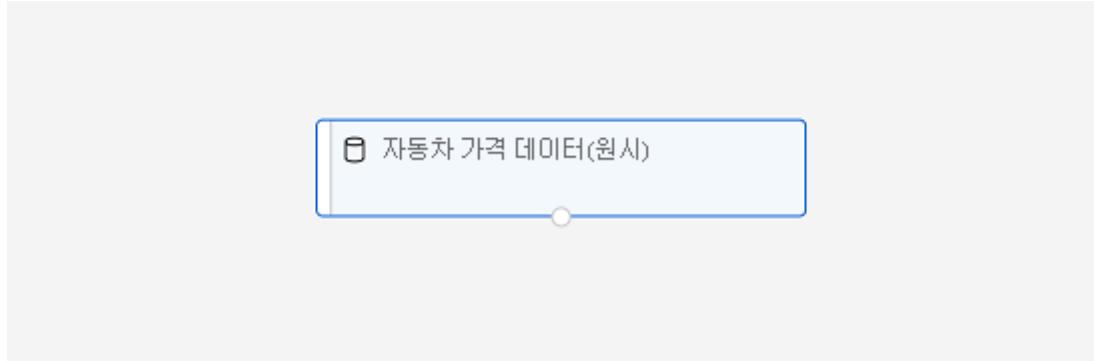
1. [Azure Machine Learning Studio](#)에서 작성 아래에 있는 디자이너 페이지를 확인하고 + 를 선택하여 새 파이프라인을 만듭니다.
2. 설정 창에서 기본 파이프라인 이름(*Pipeline-Created-on- date***)을 자동차 가격 학습 으로 변경합니다(설정 창이 보이지 않으면 상단 파이프라인 이름 옆에 있는 ☰ 아이콘을 클릭).
3. 파이프라인을 실행할 컴퓨팅 대상을 지정해야 합니다. 설정 창에서 컴퓨팅 대상 선택 을 사용하여 이전에 만든 컴퓨팅 클러스터를 선택합니다.

데이터 세트 추가 및 탐색

이 모듈에서는 특성에 따라 자동차 가격을 예측하는 회귀 모델을 학습합니다. Azure Machine Learning에는 해당 모델에 사용할 수 있는 샘플 데이터 세트가 포함되어 있습니다.

1. 디자이너 왼쪽에서 샘플 데이터 세트 섹션을 펼치고 샘플 섹션에서 자동차 가격 데이터 (원시) 데이터 세트를 캔버스에 끌어다 놓습니다.
2. 캔버스에서 자동차 가격 데이터(원시) 데이터 세트를 마우스 오른쪽 단추로 클릭(Mac에서 는 Ctrl+클릭)하고 시작화 메뉴에서 데이터 세트 출력 을 선택합니다.
3. 데이터의 스키마를 검토하여 다양한 열의 분포를 히스토그램으로 확인할 수 있습니다.
4. 가격 열이 표시될 때까지 데이터 세트의 오른쪽으로 스크롤합니다. 해당 열이 모델에서 예측할 레이블입니다.
5. 가격 열의 열 헤더를 선택하고 오른쪽 창에 표시되는 세부 정보를 확인합니다. 여기에는 열 값에 대한 다양한 통계와 열 값의 분포를 보여 주는 히스토그램이 포함됩니다.
6. 다시 왼쪽으로 스크롤하고 정규화된 손실 열 헤더를 선택합니다. 그런 다음 해당 열의 통계를 검토하여 상당 수의 누락된 값이 있는지 확인합니다. 누락된 값이 많으면 가격 레이블 예측에 유용하지 않으므로 학습에서 제외하는 것이 좋습니다.
7. 보어, 스트로크 및 마력 열의 통계에서 누락된 값의 개수를 확인합니다. 해당 열은 정규화된 손실 보다 누락된 값이 훨씬 적으므로, 값이 누락된 행을 학습에서 제외하더라도 가격을 예측하는 데 여전히 유용합니다.

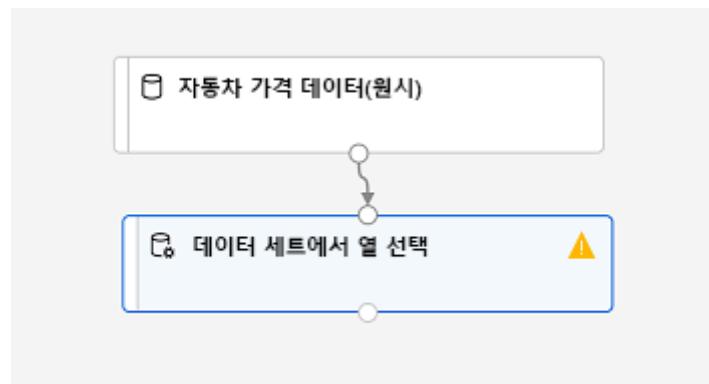
8. **스트로크, 최고 RPM 및 시내 주행 연비** 열의 값을 비교합니다. 관련 값이 모두 다른 기준으로 측정되었으므로 **최고 RPM** 의 큰 값으로 인해 학습 알고리즘이 편중될 수 있으며, **스트로크** 와 같이 값이 더 작은 열보다 해당 열에 과도한 의존도가 발생할 수 있습니다. 일반적으로 데이터 과학자는 숫자 열을 비슷한 기준에 기초하도록 '정규화'하여 해당 편중을 최소화합니다.
9. 다음과 같이 캔버스에서 데이터 세트를 볼 수 있도록 **자동차 가격 데이터(원시)** 결과 시각화 창을 닫습니다.



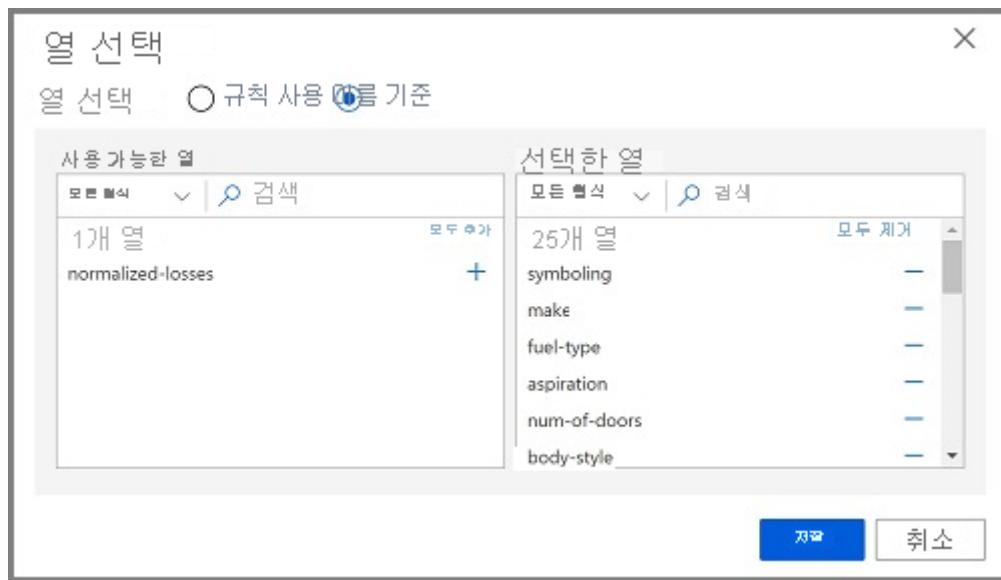
데이터 변환 추가

일반적으로 데이터 변환을 적용하여 모델링용 데이터를 준비합니다. 자동차 가격 데이터의 경우 데이터를 탐색할 때 발견된 문제를 해결하기 위해 변환을 추가합니다.

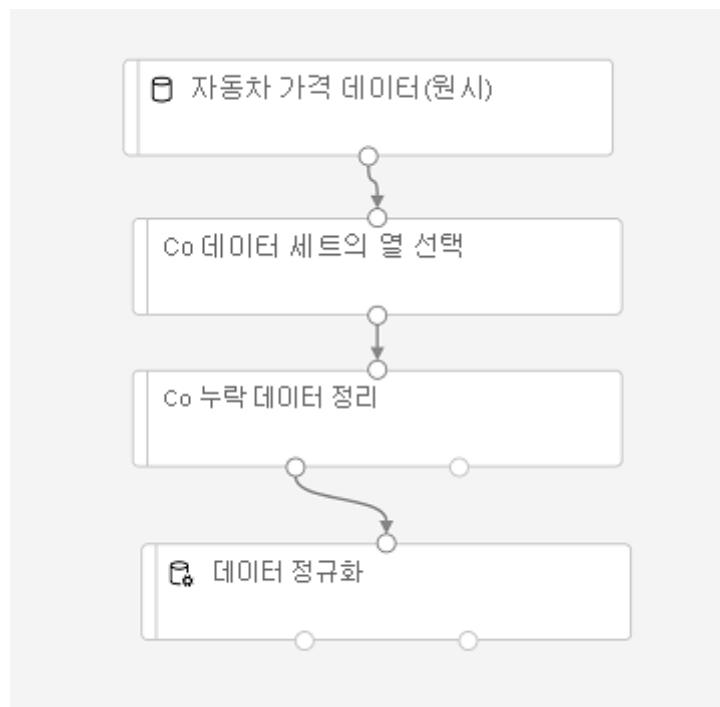
1. 왼쪽 창에서 **데이터 변환** 섹션을 펼칩니다. 여기에는 모델 학습 전에 데이터 변환에 사용 할 수 있는 다양한 모듈이 포함되어 있습니다.
2. **데이터 세트에서 열 선택** 모듈을 자동차 가격 데이터(원시) 모듈 아래의 캔버스에 끌어다 놓습니다. 그런 다음 다음과 같이 **자동차 가격 데이터(원시)** 모듈 하단의 출력을 **데이터 세트에서 열 선택** 모듈 상단의 입력에 연결합니다.



3. **데이터 세트에서 열 선택** 모듈을 선택하고 오른쪽에 있는 **설정** 창에서 **열 편집** 을 선택합니다. 그리고 나서 다음과 같이 **열 선택** 창에서 **이름순** 을 선택하고 + 링크를 사용하여 **정규화된 손실** 을 제외한 모든 열을 추가합니다.

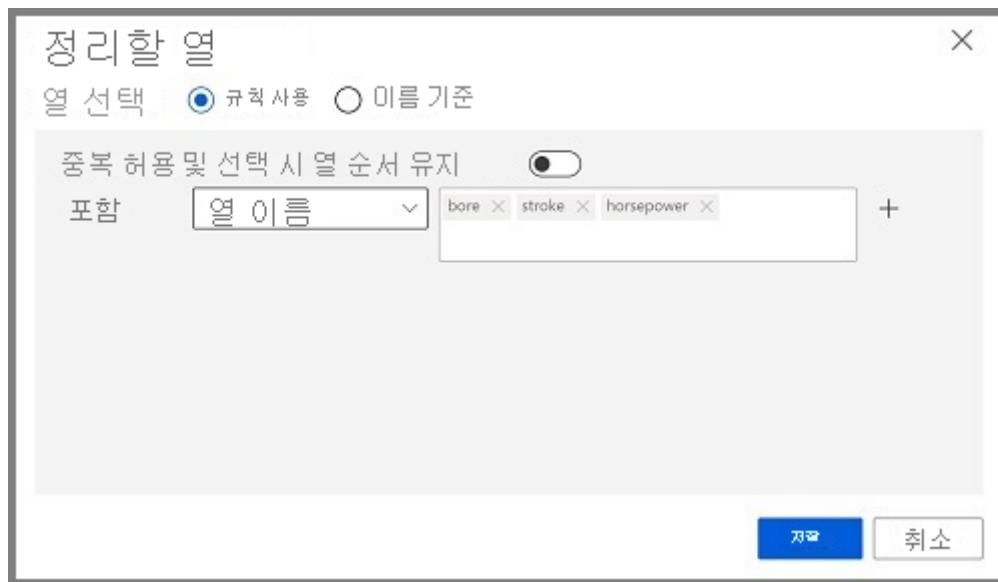


이 연습의 나머지 부분에서는 다음과 같은 파이프라인을 만듭니다.



필요한 모듈을 추가하고 구성할 때 위의 이미지를 참조하여 나머지 단계를 수행합니다.

4. 데이터 변환 섹션에서 **누락된 데이터 정리** 모듈을 데이터 세트에서 **열 선택** 모듈 아래에 끌어다 놓습니다. 그런 다음 데이터 세트에서 **열 선택** 모듈의 출력을 **누락된 데이터 정리** 모듈의 입력에 연결합니다.
5. **누락된 데이터 정리** 모듈을 선택하고 오른쪽의 설정 창에서 **열 편집**을 클릭합니다. 그리고 나서 다음과 같이 **열 선택** 창에서 **규칙 사용**을 선택하고 포함 목록에서 **열 이름**을 선택한 다음 열 이름 상자에 **보어, 스트로크 및 마력**을 입력합니다(철자와 대문자 표시가 정확히 일치해야 함).



6. 누락된 데이터 정리 모듈을 선택한 상태로 설정 창에서 다음 구성을 설정합니다.

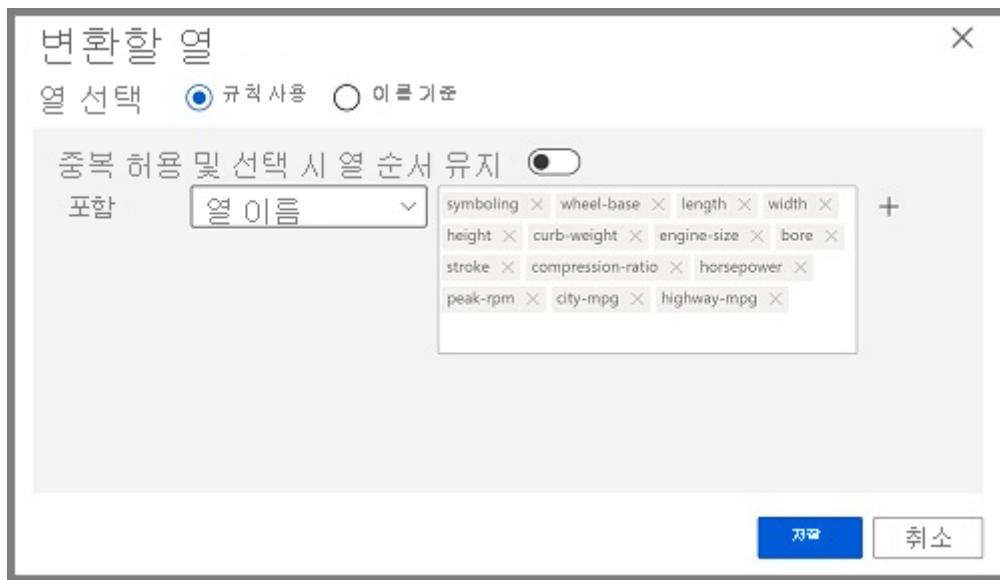
- **최소 누락 값 비율:** 0.0
- **최대 누락 값 비율:** 1.0
- **정리 모드:** 전체 행 제거

7. 데이터 정규화 모듈을 누락된 데이터 정리 모듈 아래에 있는 캔버스로 끌어다 놓습니다.

그런 다음 누락된 데이터 정리 모듈의 가장 왼쪽에 있는 출력과 데이터 정규화 모듈의 입력을 연결합니다.

8. 데이터 정규화 모듈을 선택하고 설정을 확인합니다. 여기서 변환 방법 및 변환할 열을 지정해야 한다는 것을 알 수 있습니다. 그런 다음 변환을 **MinMax**로 설정하고 다음 열 이름을 포함하도록 규칙을 적용하여 열을 편집합니다(철자, 대문자 표시 및 하이픈 넣기가 정확히 일치하는지 확인).

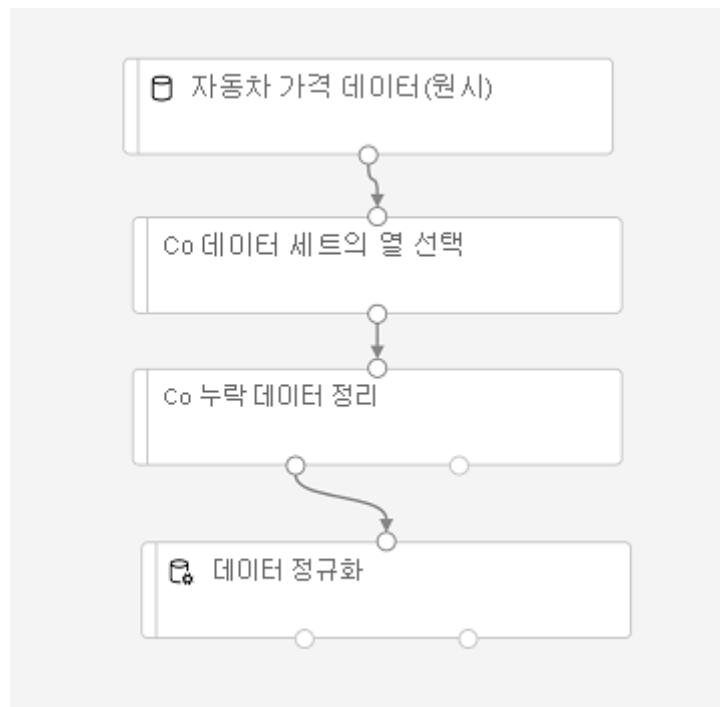
- symboling(수치 변환 값)
- wheel-base
- length
- width
- height(높이)
- curb-weight(정비 중량)
- engine-size
- bore(보어)
- stroke(스트로크)
- compression-ratio(압축비)
- horsepower(마력)
- peak-rpm
- city-mpg(시내 주행 연비)
- highway-mpg



파이프라인 실행

데이터 변환을 적용하려면 파이프라인을 실험으로 실행해야 합니다.

1. 파이프라인이 다음과 유사해야 합니다.



2. **제출** 을 선택하고 컴퓨팅 클러스터에서 **mslearn-auto-training** 이라는 새 실험으로 파이프라인을 실행합니다.
3. 실행이 끝날 때까지 기다립니다. 5분 이상 걸릴 수 있습니다. 실행이 완료되면 모듈은 다음과 같습니다.



변환된 데이터 보기

이제 모델 학습을 위한 데이터 세트가 준비되었습니다.

1. 완료된 데이터 정규화 모듈을 선택하고 오른쪽 설정 창의 출력 + 로그 탭에서 **변환된 데이터 세트**에 대한 **시각화** 아이콘을 선택합니다.
2. 데이터를 보고 **정규화된 손실** 열이 제거된 것을 확인하고, 모든 행에 **보어, 스트로크** 및 **마력** 데이터가 포함되어 있으며 선택한 숫자 열이 공통 기준으로 정규화된 것을 확인합니다.
3. 정규화된 데이터의 결과 시각화를 닫습니다.

다음 단원: 학습 파이프라인 만들기 및 실행

[계속 >](#)

[〈 이전](#)

단위 5/10 ▼

[다음 〉](#)

✓ 100 XP



학습 파이프라인 만들기 및 실행

12분

데이터 변환을 사용하여 데이터를 준비한 후에는 이를 사용하여 기계 학습 모델을 학습할 수 있습니다.

학습 모듈 추가

모델을 학습하는 데 사용하는 데이터와, 학습된 모델을 테스트하는 데 사용할 데이터를 서로 구분하는 것이 일반적입니다. 이렇게 하면 모델에서 예측하는 레이블을 원래 데이터 세트의 실제 알려진 레이블과 비교할 수 있습니다.

이 연습에서는 다음과 같이 **자동차 가격 학습** 파이프라인을 확장합니다.



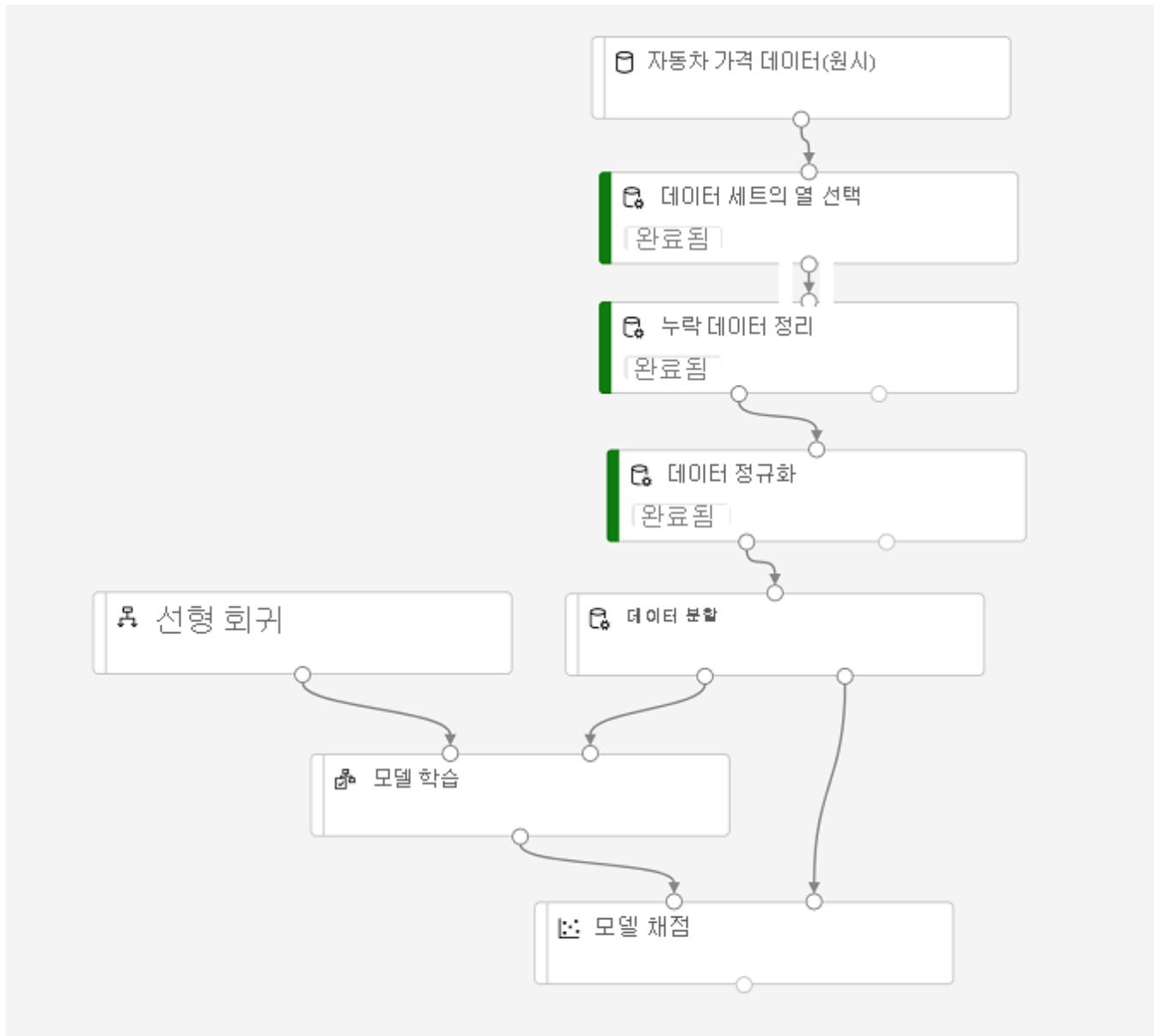
필요한 모듈을 추가하고 구성하는 동안 위 정보를 참조로 사용하여 아래의 단계를 따릅니다.

1. 아직 열려 있지 않다면 이전 단원에서 만든 **자동차 가격 학습** 파이프라인을 엽니다.
2. 왼쪽 창의 **데이터 변환** 섹션에서 **데이터 분할** 모듈을 **데이터 정규화** 모듈의 아래에 있는 캔버스로 끌어 놓습니다. 그런 다음 **데이터 정규화** 모듈의 '변환된 데이터 세트'(왼쪽) 출력을 **데이터 분할** 모듈의 입력에 연결합니다.
3. **데이터 분할** 모듈을 선택하고, 다음과 같이 설정을 구성합니다.
 - **분할 모드:** 행 분할
 - **첫 번째 출력 데이터 세트에서 행의 비율:** 0.7
 - **무작위 초기값:** 123
 - **계층화된 분할:** 아니요
4. 왼쪽 창에서 **모델 학습** 섹션을 확장하고 **데이터 분할** 모듈 아래에 있는 캔버스로 **모델 학습** 모듈을 끌어다 놓습니다. 그런 다음 **데이터 분할** 모듈의 *Result dataset1*(왼쪽) 출력을 **모델 학습** 모듈의 '데이터 세트'(오른쪽) 입력에 연결합니다.
5. 학습하고 있는 모델은 **가격** 값을 예측하므로 **모델 학습** 모듈을 선택하고 해당 설정을 수정하여 **레이블 열을 가격**으로 설정합니다(대소문자 표시 및 철자가 정확히 일치해야 함).
6. 모델에서 예측하는 **가격** 레이블은 숫자 값이므로 '회귀' 알고리즘을 사용하여 모델을 학습해야 합니다. **기계 학습** 알고리즘 섹션을 펼치고 **회귀**에서 **선형 회귀** 모듈을 **데이터 분할** 모듈의 왼쪽과 **모델 학습** 모듈의 위에 있는 캔버스로 끌어 놓습니다. 그런 다음 출력을 **모델 학습** 모듈의 **학습되지 않은 모델**(왼쪽) 입력에 연결합니다.

① 참고

회귀 모델을 학습하는 데 사용할 수 있는 여러 알고리즘이 있습니다. 이를 선택하는 데 도움이 필요하면 [Azure Machine Learning 디자이너용 기계 학습 알고리즘 치트 시트](#)를 살펴보세요.

7. 학습된 모델을 테스트하기 위해, 원래 데이터를 분할할 때 따로 분리해 두었던 유효성 검사 데이터 세트를 '채점'합니다. 즉, 유효성 검사 데이터 세트의 특징에 대해 레이블을 예측합니다. **모델 채점 및 평가** 섹션을 펼치고 **모델 채점** 모듈을 **모델 학습** 모듈 아래에 있는 캔버스로 끌어다 놓습니다. 그런 다음 **모델 학습** 모듈의 출력을 **모델 채점** 모듈의 **학습된 모델**(왼쪽) 입력에 연결하고 **데이터 분할** 모듈의 **결과 데이터 세트2**(오른쪽) 출력을 **모델 채점** 모듈의 **데이터 세트**(오른쪽) 입력에 끌어다 놓습니다.
8. 파이프라인이 다음과 같아야 합니다.



학습 파이프라인 실행

이제 학습 파이프라인을 실행하고 모델을 학습할 준비가 되었습니다.

- 제출을 선택하고 `mslearn-auto-training`이라는 기존 실험을 사용하여 파이프라인을 실행합니다.
- 실험이 완료될 때까지 기다립니다. 5분 이상 걸릴 수 있습니다.
- 실험이 완료되면 모델 채점 모듈을 선택하고 설정 창에서 출력 + 로그 탭의 데이터 출력 아래에 있는 점수가 매겨진 데이터 세트 섹션에서 시각화 아이콘을 사용하여 결과를 확인합니다.
- 오른쪽으로 스크롤하면 가격 열(레이블의 알려진 실제 값이 포함됨) 옆에 예측 레이블 값 을 포함하는 점수가 매겨진 레이블이라는 새 열이 있습니다.
- 모델 채점 결과 시각화 창을 닫습니다.

모델에서 가격 레이블의 값을 예측하지만 그 예측은 얼마나 신뢰할 만한 것일까요? 이를 평가 하려면 모델을 평가해야 합니다.

< 이전

단위 6/10 ▼

다음 >

✓ 100 XP



회귀 모델 평가

5분

회귀 모델을 평가하기 위해 학습 중에 다시 사용할 유효성 검사 데이터 세트의 실제 레이블과 예측 레이블을 비교할 수 있지만, 이는 정확하지 않은 프로세스이며 여러 모델의 성능을 비교하는 데 사용할 수 있는 간단한 메트릭을 제공하지 않습니다.

모델 평가 모듈 추가

1. 아직 열려 있지 않다면 이전 단원에서 만든 **자동차 가격 학습** 파이프라인을 엽니다.
2. 왼쪽 창의 **모델 채점 및 평가** 섹션에서 **모델 평가** 모듈을 **모델 채점** 모듈 아래에 있는 캔버스로 끌어다 놓고, **모델 채점** 모듈의 출력을 **모델 평가** 모듈의 점수가 매겨진 데이터 세트(왼쪽) 입력에 연결합니다.
3. 파이프라인이 다음과 같아야 합니다.



4. **제출** 을 선택하고 **mslearn-auto-training** 이라는 기존 실험을 사용하여 파이프라인을 실행합니다.
5. 실험이 완료될 때까지 기다립니다.
6. 실험이 완료되면 **모델 평가** 모듈을 선택하고 설정 창에서 **출력 + 로그** 탭의 **데이터 출력** 아래에 있는 **평가 결과** 섹션에서 **시각화** 아이콘을 사용하여 결과를 확인합니다. 여기에는 다음과 같은 회귀 성능 메트릭이 포함됩니다.

- **MAE(평균 절대 오차)**: 예측 값과 실제 값 사이의 평균 차이입니다. 해당 값은 레이블과 동일한 단위(이 경우 달러)를 기준으로 합니다. 해당 값이 낮을수록 모델의 예측 정확도가 높아집니다.
- **RMSE(평균 제곱 오차)**: 예측 값과 실제 값 사이의 평균 제곱 차이의 제곱근입니다. 결과는 레이블과 동일한 단위(달러)를 기준으로 하는 메트릭입니다. MAE(위)보다 차이가 크면 개별 오차의 분산이 크다는 것을 나타냅니다(예: 일부 오차는 매우 작고 다른 오차는 큰 경우).

- **RSE(상대 제곱 오차)**: 예측 값과 실제 값 간의 차이에 대한 제곱을 기준으로 한, 0에서 1 사이의 상대 메트릭입니다. 해당 메트릭이 0에 가까울수록 모델의 성능이 더 뛰어납니다. 해당 메트릭은 상대적이므로 레이블이 다른 단위인 모델을 비교하는 데 사용할 수 있습니다.
- **RAE(상대 절대 오차)**: 예측 값과 실제 값 간의 절대 차이에 대한 제곱을 기준으로 한, 0에서 1 사이의 상대 메트릭입니다. 해당 메트릭이 0에 가까울수록 모델의 성능이 더 뛰어납니다. RSE와 마찬가지로 해당 메트릭은 레이블의 단위가 서로 다른 모델을 비교하는 데 사용할 수 있습니다.
- **결정 계수(R^2)**: 해당 메트릭은 일반적으로 '결정 계수'라고 하며 모델이 설명하는 예측 값과 실제 값 간의 분산이 어느 정도인지 개략적으로 알려 줍니다. 해당 값이 1에 가까울수록 모델의 성능이 더 뛰어납니다.

7. 모델 평가 결과 시각화 창을 닫습니다.

다른 회귀 알고리즘을 사용해 **데이터 분할** 모듈의 동일한 출력을 두 번째 **모델 학습** 모듈(다른 알고리즘 사용) 및 두 번째 **모델 채점** 모듈에 연결하여 결과를 비교할 수 있습니다. 그런 다음 두 **모델 채점** 모듈의 출력을 동일한 **모델 평가** 모듈에 연결하여 나란히 비교할 수 있습니다.

요구 사항에 맞는 평가 메트릭을 사용하는 모델을 발견했다면, 해당 모델을 새 데이터와 함께 사용할 준비를 할 수 있습니다.

다음 단원: 유추 파이프라인 만들기

계속 >

< 이전

단위 7/10 ▼

다음 >

100 XP



유추 파이프라인 만들기

8분

모델을 학습하기 위한 파이프라인을 만들고 실행한 후에는 새 데이터에 대해 동일한 데이터 변환을 수행한 다음 학습된 모델을 사용하여 해당 특징에 따라 레이블 값을 '유추'(즉, 예측)하는 두 번째 파이프라인이 필요합니다. 이를 통해 애플리케이션에서 사용하도록 게시할 수 있는 예측 서비스의 기반이 형성됩니다.

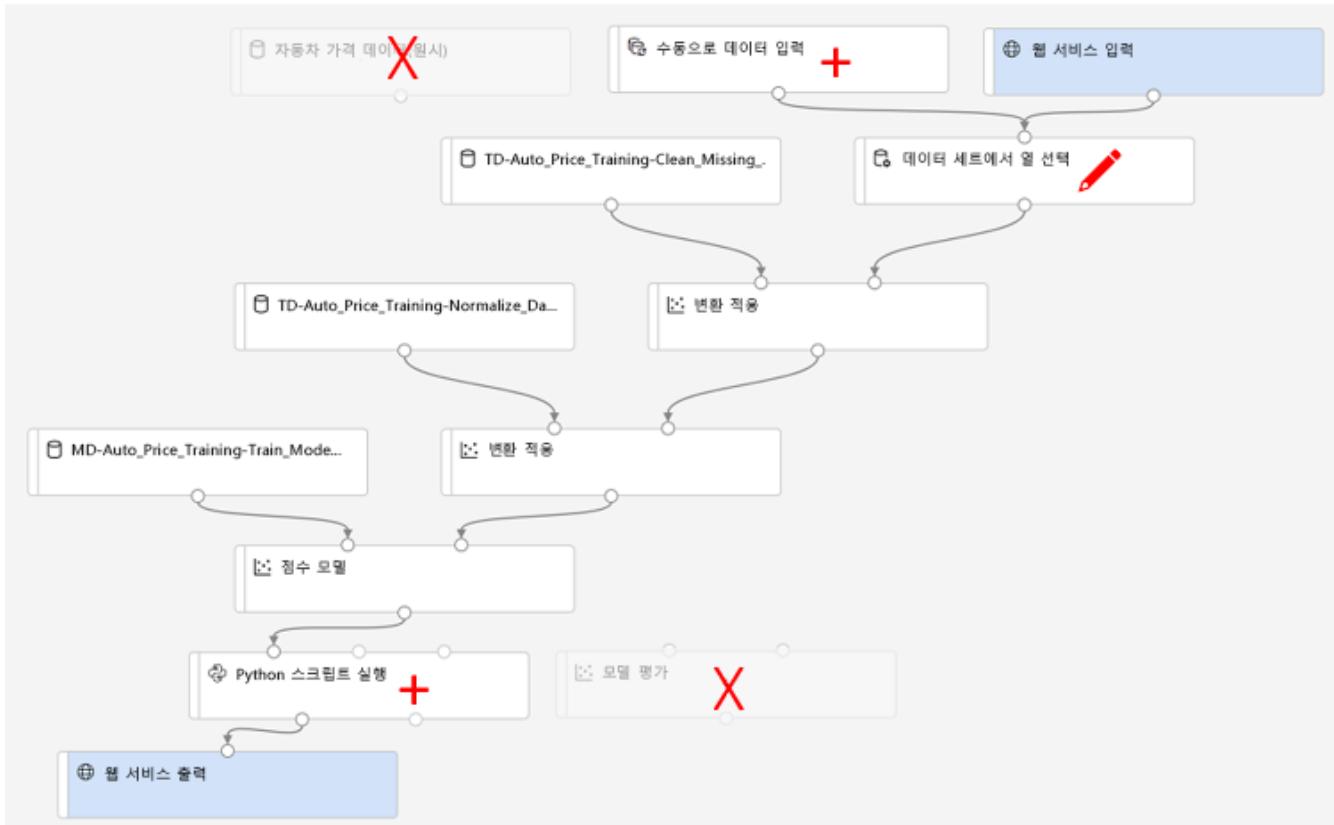
유추 파이프라인 만들기 및 실행

1. Azure Machine Learning Studio에서 **디자이너** 페이지를 클릭하여 만든 모든 파이프라인을 확인합니다. 그런 다음 이전에 만든 **자동차 가격 학습** 파이프라인을 엽니다.
2. **유추 파이프라인 만들기** 드롭다운 목록에서 **실시간 유추 파이프라인**을 클릭합니다. 몇 초 후에 **자동차 가격 학습-실시간 유추**라는 파이프라인의 새 버전이 열립니다.

파이프라인에 웹 서비스 입력과 웹 서비스 출력 모듈이 포함되지 않은 경우 **디자이너** 페이지로 돌아가 **자동차 가격 학습-실시간 유추** 파이프라인을 다시 엽니다.

3. 새 파이프라인의 이름을 **자동차 가격 예측**으로 바꾸고 새 파이프라인을 검토합니다. 여기에는 제출할 새 데이터에 대한 웹 서비스 입력과 결과를 반환하기 위한 웹 서비스 출력이 포함되어 있습니다. 학습 데이터의 통계가 새 데이터 값을 정규화하는 데 사용되고 학습된 모델이 새 데이터의 점수를 매기는 데 사용될 수 있도록 일부 변환 및 학습 단계가 해당 파이프라인에 캡슐화되었습니다.

유추 파이프라인을 다음과 같이 변경합니다.



- 자동차 가격 데이터(원시) 데이터 세트를 레이블 열(가격)을 포함하지 않은 수동으로 데이터 입력 모듈로 바꿉니다.
- 데이터 세트에서 열 선택 모듈을 수정하여 (이제 존재하지 않는) 가격 열에 대한 참조를 제거합니다.
- 모델 평가 모듈을 제거합니다.
- 예측된 레이블만 반환하도록 웹 서비스 출력 앞에 Python 스크립트 실행 모듈을 삽입합니다.

파이프라인을 수정할 때 위 이미지와 정보를 참조하여 아래의 나머지 단계를 수행 합니다.

4. 유추 파이프라인은 새 데이터가 원래 학습 데이터의 스키마와 일치하는 것으로 가정하므로 학습 파이프라인의 자동차 가격 데이터(원시) 데이터 세트가 포함됩니다. 하지만 해당 입력 데이터에는 모델이 예측하는 가격 레이블이 포함되어 있으므로, 아직 가격 예측이 이루어지지 않은 새 자동차 데이터에 이를 포함시키는 것은 비논리적입니다. 해당 모듈을 삭제하고 데이터 입력 및 출력 섹션에 있는 수동으로 데이터 입력 모듈로 바꾸어 다음 CSV 데이터를 포함시킵니다. 여기에는 세 대의 자동차에 대해 레이블이 없는 특징 값이 포함됩니다(텍스트 블록 전체를 복사하여 붙여넣기).

CSV	
-----	--

```
symboling,normalized-losses,make,fuel-type,aspiration,num-of-doors,body-style,drive-wheels,engine-location,wheel-base,length,width,height,curb-weight,engine-type,num-of-cylinders,engine-size,fuel-
```

```
system,bore,stroke,compression-ratio,horsepower,peak-rpm,city-mpg,highway-mpg
3,NaN,alfa-
romero,gas,std,two,convertible,rwd,front,88.6,168.8,64.1,48.8,2548,dohc,four,
130,mpfi,3.47,2.68,9,111,5000,21,27
3,NaN,alfa-
romero,gas,std,two,convertible,rwd,front,88.6,168.8,64.1,48.8,2548,dohc,four,
130,mpfi,3.47,2.68,9,111,5000,21,27
1,NaN,alfa-
romero,gas,std,two,hatchback,rwd,front,94.5,171.2,65.5,52.4,2823,ohcv,six,152
,mpfi,2.68,3.47,9,154,5000,19,26
```

5. 새로운 수동으로 데이터 입력 모듈을 웹 서비스 입력과 동일한 데이터 세트에서 열 선택 모듈의 데이터 세트 입력으로 연결합니다.
6. 들어오는 데이터의 스키마를 변경하여 **가격** 필드를 제외했으므로 남은 모듈에서 해당 필드를 사용할 수 없도록 제거해야 합니다. 데이터 세트에서 열 선택 모듈을 선택한 다음 설정 창에서 열을 수정하여 **가격** 필드를 제거합니다.
7. 유추 파이프라인에는 새 데이터로부터 예측하기에 유용하지 않은 **모델 평가** 모듈이 포함되어 있으므로 해당 모듈을 삭제합니다.
8. **모델 채점** 모듈의 출력에는 모든 입력 특징뿐만 아니라 예측된 레이블도 포함됩니다. 예측만 포함하도록 출력을 수정하려면 다음을 수행합니다.
 - **모델 채점** 모듈과 **웹 서비스 출력** 간의 연결을 삭제합니다.
 - Python 언어 섹션에서 **Python 스크립트 실행** 모듈을 추가하고 모든 기본 Python 스크립트를 다음 코드로 바꿉니다(점수가 매겨진 레이블 열만 선택하고 **predicted_price**로 이름 변경).

```
Python
복사

import pandas as pd

def azureml_main(dataframe1 = None, dataframe2 = None):

    scored_results = dataframe1[['Scored Labels']]
    scored_results.rename(columns={'Scored Labels':'predicted_price'},
                          inplace=True)
    return scored_results
```

Python

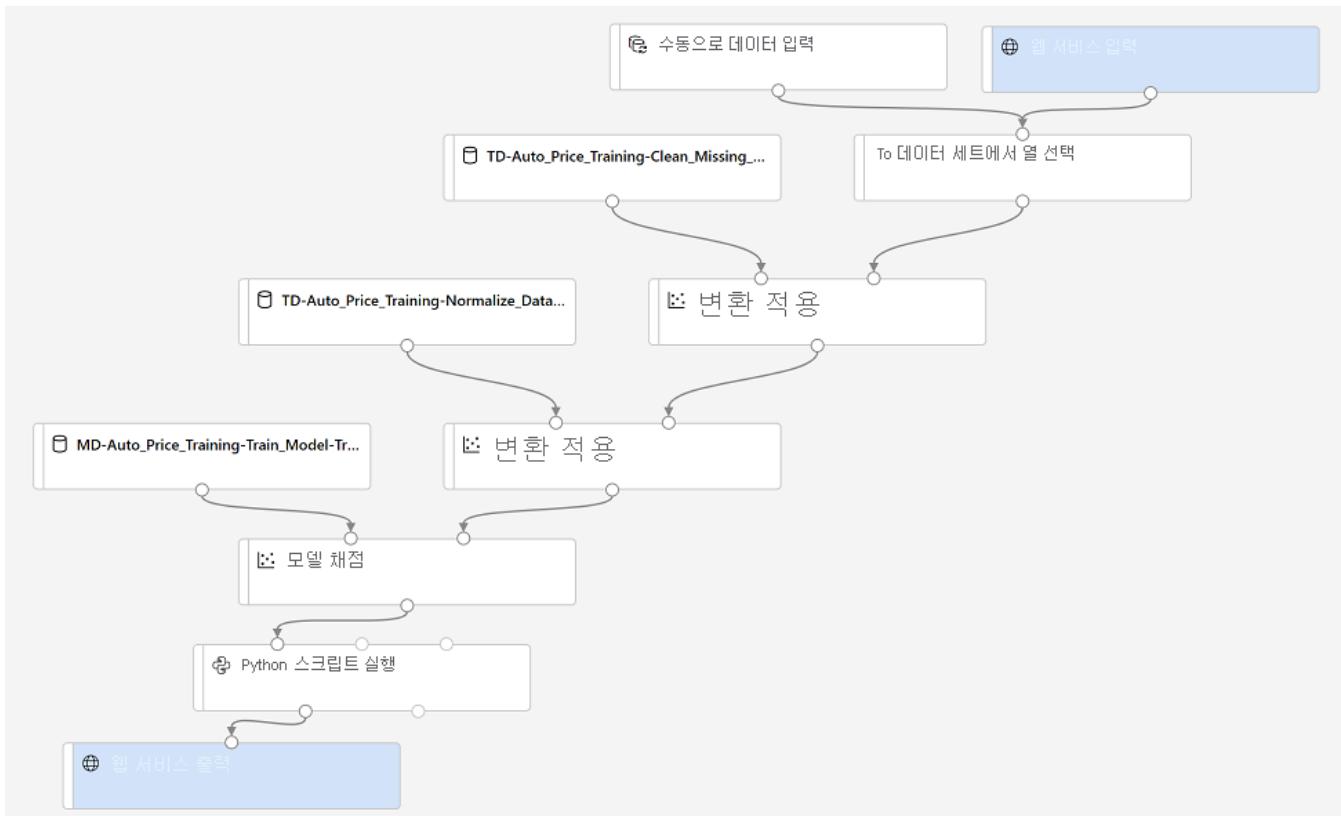
```
import pandas as pd

def azureml_main(dataframe1 = None, dataframe2 = None):

    scored_results = dataframe1[['Scored Labels']]
    scored_results.rename(columns={'Scored Labels':'predicted_price'},
                          inplace=True)
    return scored_results
```

- **모델 채점** 모듈의 출력을 **Python 스크립트 실행**의 Dataset1(맨 왼쪽) 입력에 연결하고 **Python 스크립트 실행** 모듈의 출력을 **웹 서비스 출력**에 연결합니다.

9. 파이프라인이 다음과 유사한지 확인합니다.



10. 컴퓨팅 클러스터에서 **mslearn-auto-inference** 라는 새 실험으로 파이프라인을 제출합니다. 이 작업은 다소 시간이 걸릴 수 있습니다.
11. 파이프라인이 완료되면 **Python 스크립트 실행** 모듈을 선택하고, 설정 창의 **출력 + 로그** 탭에서 **결과 데이터 세트**를 시각화하여 입력 데이터의 3개 자동차에 대한 예측 가격을 확인합니다.
12. 시각화 창을 닫습니다.

유추 파이프라인은 해당 특징에 따라 자동차 가격을 예측합니다. 이제 클라이언트 애플리케이션에서 사용할 수 있도록 파이프라인을 게시할 준비가 되었습니다.

다음 단원: 예측 서비스 배포

[계속 >](#)

[〈 이전](#)

단위 8/10 ▼

[다음 〉](#)

✓ 100 XP



예측 서비스 배포

5분

실시간 추론용 추론 파이프라인을 만들고 테스트한 후에는 이를 클라이언트 응용 프로그램에서 사용할 서비스로 게시할 수 있습니다.

① 참고

이 연습에서는 ACI(Azure Container Instance)에 웹 서비스를 배포합니다. 이러한 유형의 컴퓨팅은 동적으로 만들어지며 개발 및 테스트에 유용합니다. 프로덕션 환경에서는 '유추 클러스터'를 만들어 향상된 스케일링 성능 및 보안을 제공하는 AKS(Azure Kubernetes Service) 클러스터를 제공해야 합니다.

서비스 배포

1. 이전 단원에서 만든 **자동차 가격 예측** 유추 파이프라인을 확인합니다.
2. 오른쪽 위에서 **배포**를 선택하고 다음 설정을 사용하여 새로운 실시간 엔드포인트를 배포합니다.
 - **이름:** predict-auto-price
 - **설명:** 자동 가격 회귀
 - **컴퓨팅 형식:** Azure Container Instances
3. 웹 서비스가 배포될 때까지 기다립니다. 몇 분 정도 걸릴 수 있습니다. 배포 상태는 디자이너 인터페이스의 왼쪽 상단에 표시됩니다.

서비스 테스트

이제 클라이언트 애플리케이션에서 배포된 서비스를 테스트할 수 있습니다. 여기에서는 아래 셀의 코드를 사용하여 클라이언트 애플리케이션을 시뮬레이션합니다.

1. **엔드포인트** 페이지에서 predict-auto-price 실시간 엔드포인트를 엽니다.
2. predict-auto-price 엔드포인트가 열리면 **사용** 탭을 보고 다음 정보를 확인합니다. 클라이언트 애플리케이션에서 배포된 서비스로 연결하려면 해당 정보가 필요합니다.
 - 서비스에 대한 REST 엔드포인트
 - 서비스에 대한 기본 키

3. 해당 값 옆에 있는 링크를 사용하여 이를 클립보드에 복사할 수 있습니다.
4. 브라우저에서 **predict-auto-price** 서비스 페이지의 **사용** 페이지가 열린 상태에서 새 브라우저 탭을 열고 [Azure Machine Learning Studio](#) 의 두 번째 인스턴스를 엽니다. 그런 다음, 새 탭에서 **Notebooks** 페이지(작성자 아래)를 봅니다.
5. **Notebooks** 페이지의 **내 파일**에서 단추를 사용하여 다음 설정으로 새 파일을 만듭니다.
 - **파일 위치:** Users/사용자 이름
 - **파일 이름:** Test-Autos
 - **파일 형식:** Notebook
 - **이미 있는 경우 덮어쓰기:** 선택됨
6. 새 Notebook이 만들어지면 이전에 만든 컴퓨팅 인스턴스가 **컴퓨팅** 상자에서 선택되었는지, 그리고 상태가 **실행 중**인지 확인합니다.
7. << 단추를 사용하여 파일 탐색기 창을 축소하고 **Test-Autos.ipynb** Notebook 탭에 사용할 수 있는 공간을 확보합니다.
8. Notebook에서 만들어진 사각형 셀에 다음 코드를 붙여넣습니다.

Python

복사

```
endpoint = 'YOUR_ENDPOINT' #Replace with your endpoint
key = 'YOUR_KEY' #Replace with your key

import urllib.request
import json
import os

# Prepare the input data
data = {
    "Inputs": {
        "WebServiceInput0": [
            [
                {
                    'symboling': 3,
                    'normalized-losses': None,
                    'make': "alfa-romero",
                    'fuel-type': "gas",
                    'aspiration': "std",
                    'num-of-doors': "two",
                    'body-style': "convertible",
                    'drive-wheels': "rwd",
                    'engine-location': "front",
                    'wheel-base': 88.6,
                    'length': 168.8,
                    'width': 64.1,
                    'height': 48.8,
                    'curb-weight': 2548,
                    'engine-type': "dohc",
                    'horsepower': 95,
                    'stroke': 3.9,
                    'compression-ratio': 9.0,
                    'fuel-system': "multi-point",
                    'num-of-cylinders': "four"
                }
            ]
        ]
    }
}
```

```

        'num-of-cylinders': "four",
        'engine-size': 130,
        'fuel-system': "mpfi",
        'bore': 3.47,
        'stroke': 2.68,
        'compression-ratio': 9,
        'horsepower': 111,
        'peak-rpm': 5000,
        'city-mpg': 21,
        'highway-mpg': 27,
    },
],
},
},
"GlobalParameters": {
}
}

body = str.encode(json.dumps(data))
headers = {'Content-Type': 'application/json', 'Authorization': ('Bearer ' +
key)}
req = urllib.request.Request(endpoint, body, headers)

try:
    response = urllib.request.urlopen(req)
    result = response.read()
    json_result = json.loads(result)
    y = json_result[ "Results" ][ "WebServiceOutput0" ][ 0 ][ "predicted_price" ]
    print('Predicted price: {:.2f}'.format(y))

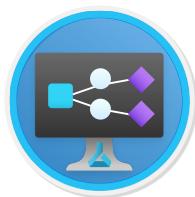
except urllib.error.HTTPError as error:
    print("The request failed with status code: " + str(error.code))

    # Print the headers to help debug the error
    print(error.info())
    print(json.loads(error.read().decode("utf8", 'ignore')))
```

① 참고

코드의 세부 정보에 대해 너무 걱정하지 마세요. 자동차의 세부 정보를 제출하고 만 들어진 predict-auto-price 서비스를 사용하여 예측 가격을 얻습니다.

9. predict-auto-price 서비스의 [사용](#) 페이지가 포함된 브라우저 탭으로 전환한 다음 서비스에 대한 REST 엔드포인트를 복사합니다. Notebook을 포함하는 탭으로 돌아가 키를 코드에 복사하고 YOUR_ENDPOINT를 대체합니다.
10. predict-auto-price 서비스의 [사용](#) 페이지가 포함된 브라우저 탭으로 전환한 다음 서비스에 대한 기본 키를 복사합니다. Notebook을 포함하는 탭으로 돌아가 키를 코드에 복사하고 YOUR_KEY를 대체합니다.
11. Notebook을 저장합니다. 셀 옆에 있는 ▶ 단추를 사용하여 코드를 실행합니다.



✓ 1100 XP ➔

Azure Machine Learning 디자이너를 사용하여 분류 모델 만들기

1 시간 • 모듈 • 10 단위

★★★★★ 4.7 (1,277)

평가하기

초급 AI 엔지니어 데이터 과학자 Machine Learning

분류는 범주 또는 클래스를 예측하는 데 사용되는 감독형 기계 학습 기술입니다. Azure Machine Learning 디자이너를 사용하여 분류 모델을 만드는 방법에 대해 알아봅니다.

학습 목표

Azure Machine Learning 디자이너를 사용하여 분류 모델 학습 및 게시

책갈피 + 컬렉션에 추가

필수 조건

Azure Portal 탐색 능력

이 모듈은 이러한 학습 경로의 일부입니다.

[Azure Machine Learning을 사용하여 코드 없는 예측 모델 만들기](#)

소개

2분



Azure Machine Learning 작업 영역 만들기

5분



컴퓨팅 리소스 만들기

5분



데이터 살펴보기

10분



학습 파이프라인 만들기 및 실행

12분



분류 모델 평가

10분



유추 파이프라인 만들기



8분

예측 서비스 배포

5분

지식 점검

2분

요약

1분

✓ 100 XP



소개

2분

분류는 항목이 속한 범주 또는 클래스를 예측하는 데 사용되는 기계 학습의 한 형태입니다. 예를 들면, 진료소는 환자가 당뇨병이 걸릴 위험이 있는지를 예측하기 위하여 환자의 특성 (나이, 체중, 혈압 등)을 이용할 수 있습니다. 이 경우 환자의 특성이 특징이며, 레이블은 각각 당뇨병이 없거나 있는 환자를 나타내는 0 또는 1이라는 분류입니다.



분류는 모델이 학습을 통해 특징 조합을 레이블에 맞추도록 레이블에 대한 특징과 알려진 값을 모두 포함하는 데이터를 사용하여 모델을 학습하는 감독형 기계 학습 기술의 예입니다. 그리고 나서, 학습이 완료된 후 학습된 모델을 사용하여 레이블이 알려지지 않은 새 항목에 대한 레이블을 예측할 수 있습니다.

Microsoft Azure Machine Learning 디자이너를 사용하여 코드를 작성할 필요 없이 끌어다 놓기 시각적 인터페이스를 사용하여 분류 모델을 만들 수 있습니다.

이 모듈에서는 다음을 수행하는 방법을 알아봅니다.

- Azure Machine Learning 디자이너를 사용하여 분류 모델을 학습합니다.
- 유추에 대한 분류 모델을 사용합니다.
- 분류 모델을 서비스로 배포합니다.

이 모듈을 완료하려면 Microsoft Azure 구독이 필요합니다. 구독이 아직 없다면 <https://azure.microsoft.com>에서 평가판에 가입할 수 있습니다.

다음 단원: Azure Machine Learning 작업 영역 만들기

계속 >

< 이전

단위 2/10 ▼

다음 >

✓ 100 XP



Azure Machine Learning 작업 영역 만들기

5분

Azure Machine Learning은 Azure에서 기계 학습 솔루션을 빌드하고 운영하기 위한 클라우드 기반 플랫폼입니다. 데이터 과학자가 데이터를 준비하고, 모델을 학습하고, 예측 서비스를 게시하고, 사용량을 모니터링하는 데 도움이 되는 여러 특징 및 기능을 포함합니다. 해당 특징 중 하나는 '디자이너'라는 시각적 인터페이스로, 코드 작성 없이 기계 학습 모델을 학습, 테스트 및 배포할 수 있게 해 줍니다.

Azure Machine Learning 작업 영역 만들기

Azure Machine Learning을 사용하려면 Azure 구독에서 '작업 영역'을 만듭니다. 그런 다음 이 작업 영역을 사용하여 데이터, 컴퓨팅 리소스, 코드, 모델 및 기계 학습 워크로드와 관련된 기타 아티팩트를 관리할 수 있습니다.

ⓘ 참고

이 모듈은 Azure Machine Learning 작업 영역을 사용하는 여러 가지 모듈 중 하나입니다.

Azure AI 기본 사항 또는 **Azure 데이터 과학자** 인증을 준비하는 과정에서 이 모듈을 진행하는 경우, 작업 영역을 한 번 만들어 두고 다른 모듈에서 재사용하는 것이 좋습니다. 각 모듈을 완료한 후에는 모듈 끝에 있는 정리 지침에 따라 컴퓨팅 리소스를 중지해야 합니다.

아직 없는 경우 다음 단계에 따라 작업 영역을 만듭니다.

1. Microsoft 자격 증명을 사용하여 [Azure Portal](#) 에 로그인합니다.
2. + 리소스 만들기를 선택하고, 'Machine Learning'을 검색하고, 다음 설정을 포함하여 새 Machine Learning 리소스를 만듭니다.

- **구독:** 'Azure 구독'
- **리소스 그룹:** '리소스 그룹 만들기 또는 선택'
- **작업 영역 이름:** '작업 영역의 고유한 이름 입력'
- **지역:** '지리적으로 가장 가까운 지역 선택'
- **스토리지 계정:** '작업 영역에 대해 만들 새로운 기본 스토리지 계정'
- **키 자격 증명 모음:** '작업 영역에 대해 만들 새로운 기본 키 자격 증명 모음'
- **Application insights:** '작업 영역에 대해 만들 새로운 기본 Application Insights 리소스'
- **컨테이너 레지스트리:** 없음('처음으로 컨테이너에 모델을 배포할 때 자동으로 만들 어짐')

3. 작업 영역이 만들어질 때까지 기다립니다(몇 분이 걸릴 수 있음). 그런 다음 포털에서 해당 위치로 이동합니다.
4. 작업 영역의 **개요** 페이지에서 Azure Machine Learning Studio를 실행(또는 새 브라우저 탭을 열고 <https://ml.azure.com> 으로 이동)한 다음 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다.
5. Azure Machine Learning Studio에서 왼쪽 상단에 있는 ☰ 아이콘을 설정/해제하면 인터페이스에 다양한 페이지가 표시됩니다. 해당 페이지를 사용하여 작업 영역에서 리소스를 관리할 수 있습니다.

Azure Portal을 사용하여 작업 영역을 관리할 수 있지만 데이터 과학자 및 기계 학습 운영 엔지니어의 경우 Azure Machine Learning Studio에서 작업 영역 리소스를 관리하는 데 특화된 사용자 인터페이스를 이용할 수 있습니다.

다음 단원: 컴퓨팅 리소스 만들기

계속 >

< 이전

단위 3/10 ▼

다음 >

✓ 100 XP



컴퓨팅 리소스 만들기

5분

Azure Machine Learning 디자이너를 사용하여 모델을 학습하고 배포하려면 학습 프로세스를 실행하고, 배포 후 학습된 모델을 테스트하기 위한 컴퓨팅이 필요합니다.

컴퓨팅 대상 만들기

컴퓨팅 대상은 모델 학습 및 데이터 탐색 프로세스를 실행할 수 있는 클라우드 기반 리소스입니다.

1. [Azure Machine Learning Studio](#)에서 관리 아래에 있는 컴퓨팅 페이지를 확인합니다. 해당 페이지는 데이터 과학 관련 활동을 위해 컴퓨팅 대상을 관리하는 곳입니다. 다음 네 가지 종류의 컴퓨팅 리소스를 만들 수 있습니다.

- **컴퓨팅 인스턴스**: 데이터 과학자가 데이터 및 모델을 작업하는 데 사용할 수 있는 개발 워크스테이션입니다.
- **컴퓨팅 클러스터**: 실험 코드의 주문형 처리를 지원하는 확장 가능한 가상 클러스터입니다.
- **유추 클러스터**: 학습된 모델을 사용하는 예측 서비스의 배포 대상입니다.
- **연결된 컴퓨팅**: Virtual Machines 또는 Azure Databricks 클러스터와 같은 기존 Azure 컴퓨팅 리소스에 연결합니다.

2. 컴퓨팅 인스턴스 탭에서 다음 설정을 사용하여 새 컴퓨팅 인스턴스를 추가합니다. 이 인스턴스를 모델을 테스트할 워크스테이션으로 사용합니다.

- **가상 머신 유형**: CPU
- **가상 머신 크기**: Standard_DS11_v2(모든 옵션에서 선택을 선택하여 이 가상 머신 크기를 검색하여 선택함)
- **컴퓨팅 이름**: 고유한 이름 입력
- **SSH 액세스 사용**: 선택 안 함

3. 컴퓨팅 인스턴스가 생성되는 동안 컴퓨팅 클러스터 탭으로 전환하고 다음 설정을 사용하여 새 컴퓨팅 클러스터를 추가합니다. 이를 사용하여 기계 학습 모델을 학습합니다.

- **가상 머신 우선 순위**: 전용
- **가상 머신 유형**: CPU
- **가상 머신 크기**: Standard_DS11_v2(모든 옵션에서 선택을 선택하여 이 가상 머신 크기를 검색하여 선택함)

- **컴퓨팅 이름:** 고유한 이름 입력
- **최소 노드 수:** 0
- **최대 노드 수:** 2
- **스케일 다운 전 유휴 시간(초):** 120
- **SSH 액세스 사용:** 선택 안 함

① 참고

이 모듈을 완료하지 않기로 결정한 경우 컴퓨팅 인스턴스를 중지하여 Azure 구독에 불필요한 요금이 발생되지 않도록 해야 합니다.

컴퓨팅 대상은 만들어지는 데 시간이 걸립니다. 대기하는 동안 다음 단원으로 이동할 수 있습니다.

다음 단원: 데이터 살펴보기

계속 >

< 이전

단위 4/10 ▼

다음 >

✓ 100 XP



데이터 살펴보기

10분

분류 모델을 학습하려면, 기존 특징(예측의 대상이 되는 엔터티의 특성)과 알려진 레이블 값(모델을 학습해 예측할 클래스 지표)이 포함된 데이터 세트가 필요합니다.

데이터 세트 만들기

Azure Machine Learning에서 모델 학습 및 기타 작업의 데이터는 주로 데이터 세트라는 개체에 캡슐화됩니다.

1. [Azure Machine Learning Studio](#)에서 데이터 세트 페이지를 확인합니다. 데이터 세트는 Azure ML에서 사용할 특정 데이터 파일이나 테이블을 나타냅니다.
2. 다음 설정을 사용하여 웹 파일에서 데이터 세트를 만듭니다.

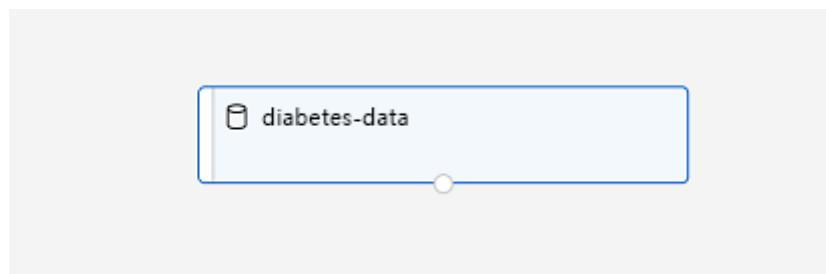
- **기본 정보:**
 - 웹 URL: <https://aka.ms/diabetes-data>
 - 이름: diabetes-data
 - 데이터 세트 형식: 테이블 형식
 - 설명: 당뇨병 데이터
- **설정 및 미리 보기:**
 - 파일 형식: 구분 기호로 분리됨
 - 구분 기호: 쉼표
 - 인코딩: UTF-8
 - 열 머리글: 첫 번째 파일의 머리글 사용
 - 행 건너뛰기: 없음
- **스키마:**
 - 경로 이외의 모든 열 포함
 - 자동으로 검색된 형식 검토
- **세부 정보 확인:**
 - 만든 후 데이터 세트를 프로파일링하지 않음

3. 데이터 세트를 만든 후에는 이를 열고 **탐색** 페이지를 보면서 데이터 샘플을 확인합니다. 이 데이터는 당뇨병에 대한 테스트를 받은 환자의 세부 정보를 나타냅니다.

파이프라인 만들기

Azure Machine Learning 디자이너는 시작하려면 먼저 파이프라인을 만들고 작업하려는 데이터 세트를 추가해야 합니다.

1. 작업 영역의 [Azure Machine Learning Studio](#)에서 **디자이너** 페이지를 보고 +를 선택하여 새 파이프라인을 만듭니다.
2. **설정** 창에서 기본 파이프라인 이름(*Pipeline-Created-on- date***)을 **Diabetes Training**으로 변경합니다(설정 창이 표시되지 않은 경우 상단 파이프라인 이름 옆에 있는 ☰ 아이콘을 클릭).
3. 파이프라인을 실행할 컴퓨팅 대상을 지정해야 합니다. **설정** 창에서 **컴퓨팅 대상 선택**을 클릭하고 이전에 만든 **aml-cluster** 컴퓨팅 클러스터를 선택합니다.
4. 디자이너의 왼쪽에서 **데이터 세트** 섹션을 확장한 다음 이전 연습에서 만든 **diabetes-data** 데이터 세트를 캔버스에 끌어다 놓습니다.
5. 캔버스에 있는 **diabetes-data** 데이터 세트를 마우스 오른쪽 단추로 클릭(Mac의 경우 Ctrl+클릭)하고 **시각화** 메뉴에서 **데이터 세트 출력**을 선택합니다.
6. 데이터의 스키마를 검토하여 다양한 열의 분포를 히스토그램으로 확인할 수 있습니다.
7. 오른쪽으로 스크롤하여 **Diabetic** 열의 열 머리글을 선택하고 0과 1, 두 값을 포함하고 있는지 확인합니다. 이러한 값은 모델에서 예측하는 레이블에 대해 가능한 두 가지 클래스를 나타내며, 0 값은 환자가 당뇨병이 아니라는 의미, 1 값은 당뇨병이라는 의미입니다.
8. 왼쪽으로 다시 스크롤하고 다른 열을 검토합니다. 이 열은 레이블을 예측하는 데 사용되는 특징을 나타냅니다. 이러한 열의 대부분은 숫자이지만 기능마다 크기가 다릅니다. 예를 들어 **Age** 값의 범위는 21~77인 반면, **DiabetesPedigree** 값의 범위는 0.078~2.3016입니다. 기계 학습 모델을 학습할 때 큰 값에 결과 예측 함수가 좌지우지되어서 작은 규모에 대한 특징의 영향을 줄이는 경우가 간혹 있을 수 있습니다. 일반적으로 데이터 과학자는 숫자 열을 비슷한 기준에 기초하도록 정규화하여 편중을 최소화합니다.
9. 다음과 같이 캔버스에서 데이터 세트를 볼 수 있도록 **diabetes-data**의 **결과 시각화** 창을 닫습니다.

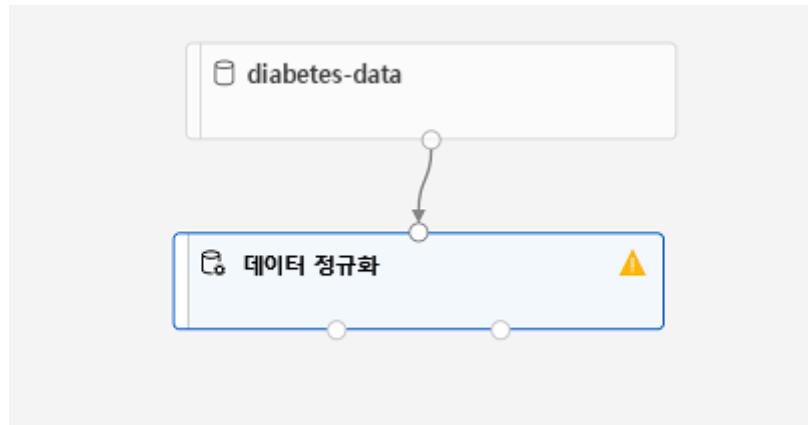


변환 추가

모델을 학습하려면 보통 데이터에 일부 전처리 변환을 적용해야 합니다.

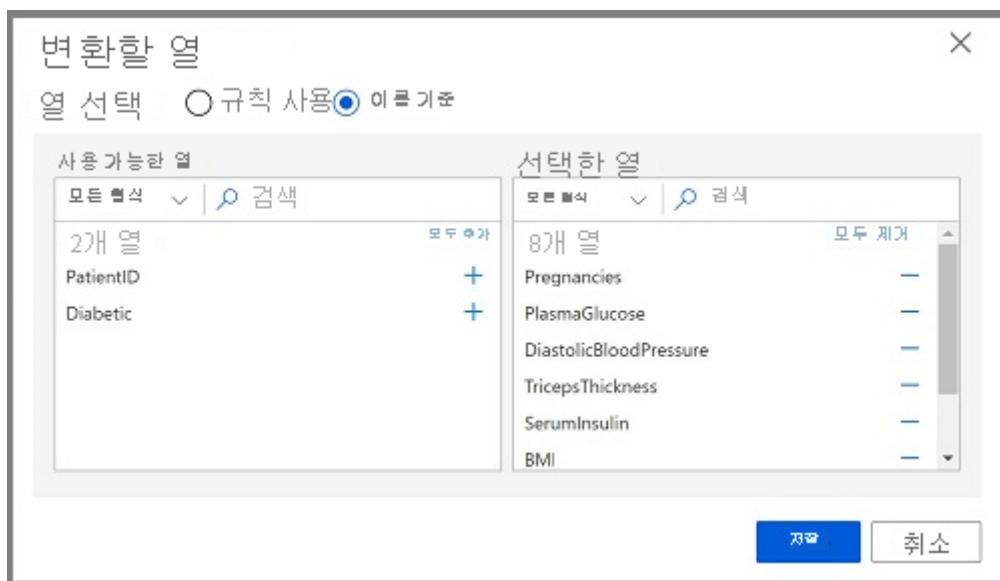
1. 왼쪽 창에서 **데이터 변환** 섹션을 펼칩니다. 여기에는 모델 학습 전에 데이터 변환에 사용할 수 있는 다양한 모듈이 포함되어 있습니다.
2. **데이터 정규화** 모듈을 **diabetes-data** 데이터 세트 아래에 있는 캔버스로 끌어다 놓습니다. 그런 다음 **diabetes-data** 데이터 세트 하단의 출력을 다음과 같이 **데이터 정규화** 모듈 상

단의 입력에 연결합니다.



3. 데이터 정규화 모듈을 선택하고 설정을 확인합니다. 여기서 변환 방법 및 변환할 열을 지정해야 한다는 것을 알 수 있습니다.
4. 이미지에 표시된 것처럼 변환을 MinMax로 설정하고 다음 열을 이름으로 포함하도록 열을 편집합니다.

- Pregnancies
- PlasmaGlucose
- DiastolicBloodPressure
- TricepsThickness
- SerumInsulin
- BMI
- DiabetesPedigree
- Age



데이터 변환은 숫자 열을 정규화하여 동일한 규모로 배치하므로, 큰 값이 있는 열이 모델 학습을 좌지우지하지 않도록 합니다. 일반적으로 학습을 위한 데이터를 준비하려고 이와 같은 일련의 사전 처리 변환을 적용하지만 이 연습에서는 작업을 간단하게 유지합니다.

파이프라인 실행

데이터 변환을 적용하려면 파이프라인을 실험으로 실행해야 합니다.

1. 파이프라인이 다음과 유사해야 합니다.



2. **제출** 을 선택하고 컴퓨팅 클러스터에서 **mslearn-diabetes-training** 이라는 새 실험으로 파이프라인을 실행합니다.
3. 실행이 완료될 때까지 기다려 주세요. 몇 분 정도 걸릴 수 있습니다.

변환된 데이터 보기

이제 모델 학습을 위한 데이터 세트가 준비되었습니다.

1. 완료된 데이터 정규화 모듈을 선택하고 오른쪽 설정 창의 출력 + 로그 탭에서 **변환된 데이터 세트**에 대한 시각화 아이콘을 선택합니다.
2. 데이터를 보고 선택한 숫자 열이 공통 배율로 정규화된 것을 확인합니다.
3. 정규화된 데이터의 결과 시각화를 닫습니다.

다음 단원: 학습 파이프라인 만들기 및 실행

계속 >

< 이전

단위 5/10 ▼

다음 >

✓ 100 XP



학습 파이프라인 만들기 및 실행

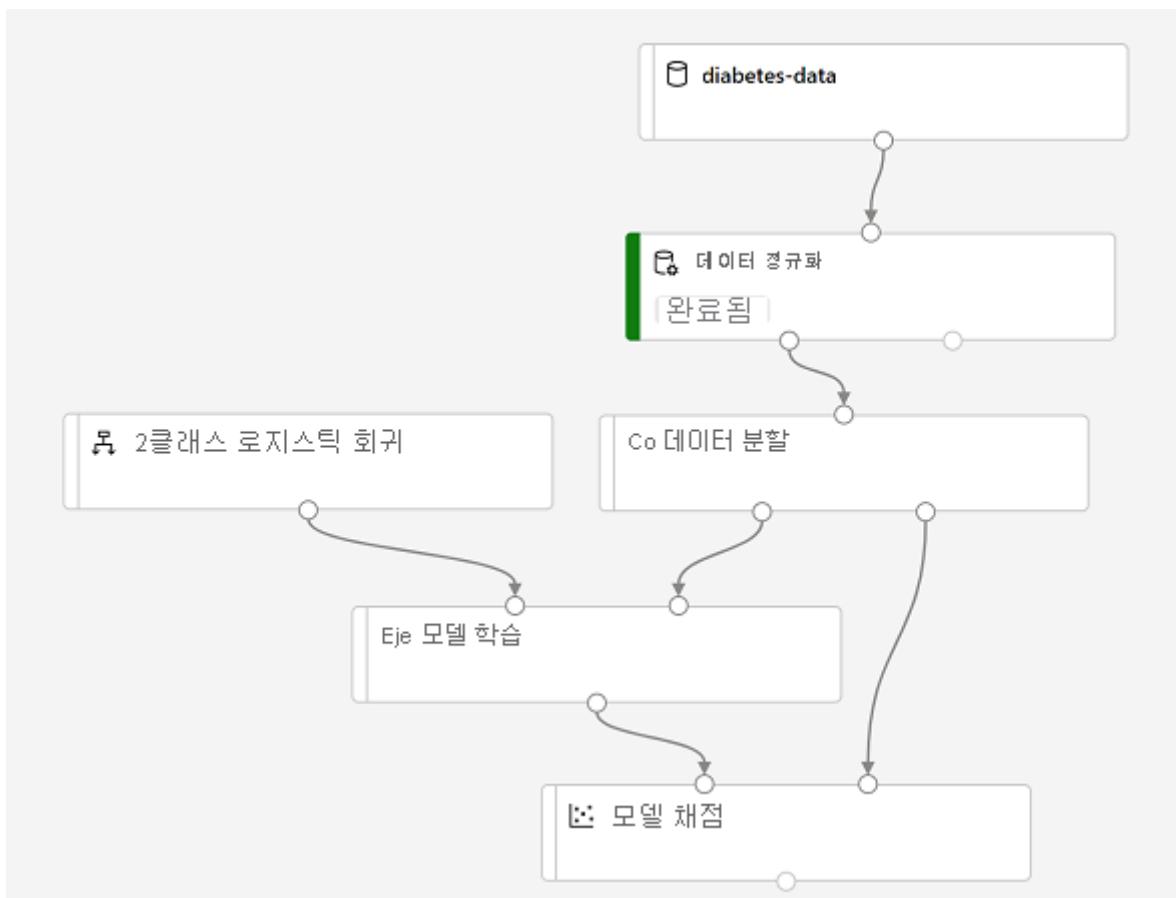
12분

데이터 변환을 사용하여 데이터를 준비한 후에는 이를 사용하여 기계 학습 모델을 학습할 수 있습니다.

학습 모듈 추가

모델을 학습하는 데 사용하는 데이터와, 학습된 모델을 테스트하는 데 사용할 데이터를 서로 구분하는 것이 일반적입니다. 이렇게 하면 모델에서 예측하는 레이블을 원래 데이터 세트의 실제 알려진 레이블과 비교할 수 있습니다.

이 연습에서는 다음과 같이 **당뇨병 학습** 파이프라인을 확장합니다.



아래 단계를 따르고, 필요한 모듈을 추가하고 구성할 때 위 이미지를 참조용으로 사용합니다.

1. 아직 열려 있지 않다면 이전 단원에서 만든 **당뇨병 학습** 파이프라인을 엽니다.
2. 왼쪽 창의 **데이터 변환** 섹션에서 **데이터 분할** 모듈을 **데이터 정규화** 모듈의 아래에 있는 캔버스로 끌어 놓습니다. 그런 다음 **데이터 정규화** 모듈의 '변환된 데이터 세트'(왼쪽) 출력을 **데이터 분할** 모듈의 입력에 연결합니다.

3. 데이터 분할 모듈을 선택하고, 다음과 같이 설정을 구성합니다.

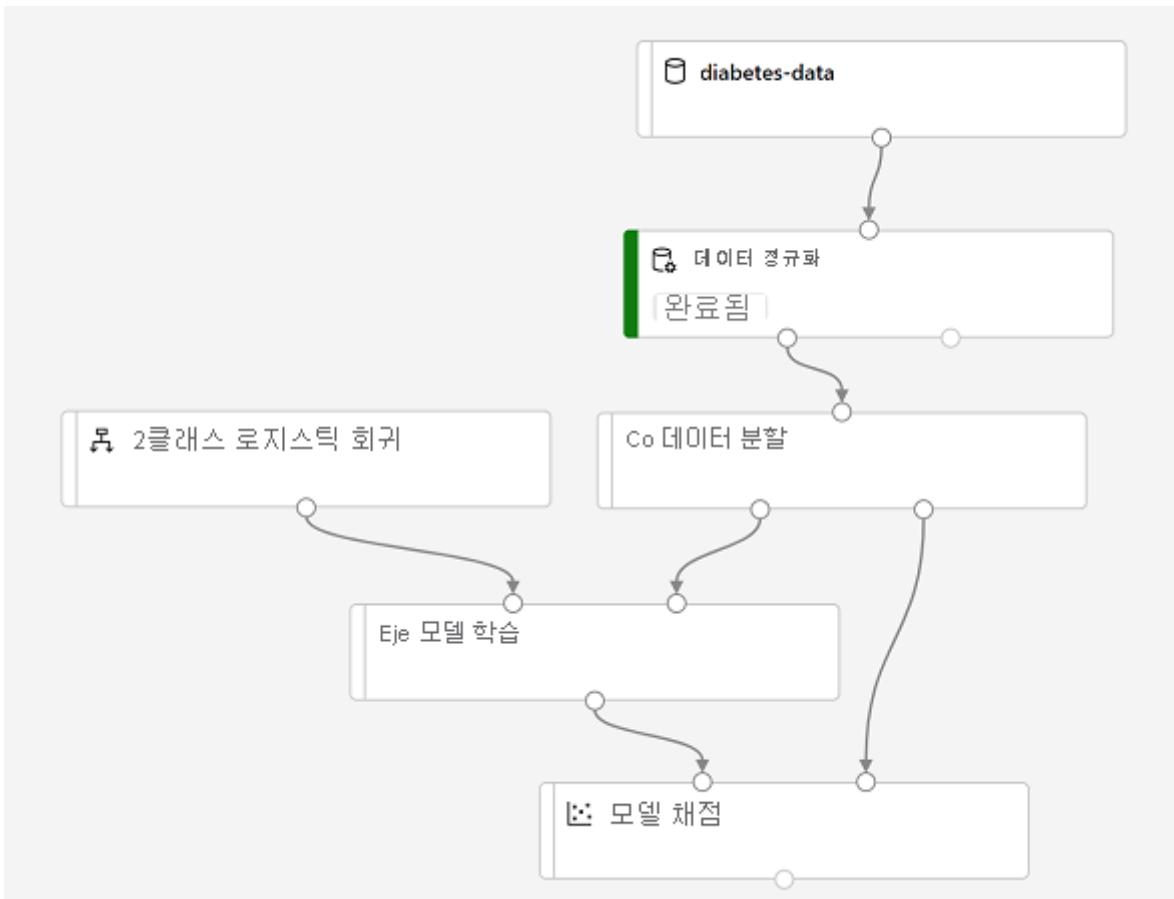
- **분할 모드** 행 분할
- 첫 번째 출력 데이터 세트의 행의 비율: 0.7
- 무작위 초기값: 123
- 계층화된 분할: 아닙요

4. 왼쪽 창에서 **모델 학습** 섹션을 확장하고 **데이터 분할** 모듈 아래에 있는 캔버스로 **모델 학습** 모듈을 끌어다 놓습니다. 그런 다음 **분할 데이터** 모듈의 Result dataset1(왼쪽) 출력을 **모델 학습** 모듈의 데이터 세트(오른쪽) 입력에 연결합니다.
5. 학습하고 있는 모델은 **당뇨** 값을 예측하므로 **모델 학습** 모듈을 선택하고 해당 설정을 수정하여 **레이블 열**을 Diabetic(당뇨)으로 설정합니다(대문자 표시 및 철자 정확히 일치).
6. 모델에서 예측하는 Diabetic 레이블은 클래스(0 또는 1)이므로 분류 알고리즘을 사용하여 모델을 학습해야 합니다. 특히 가능한 클래스가 두 가지이므로 이진 분류 알고리즘이 필요합니다. **기계 학습 알고리즘** 섹션을 펼치고 **분류**에서 **2 클래스 로지스틱 회귀 분석** 모듈을 **데이터 분할** 모듈의 왼쪽, **모델 학습** 모듈 위의 캔버스로 끌어 놓습니다. 그런 다음 출력을 **모델 학습** 모듈의 **학습되지 않은 모델**(왼쪽) 입력에 연결합니다.

① 참고

분류 모델을 학습하는 데 사용할 수 있는 여러 알고리즘이 있습니다. 이를 선택하는 데 도움이 필요하면 [Azure Machine Learning 디자이너용 기계 학습 치트 시트](#)를 살펴보세요.

7. 학습된 모델을 테스트하기 위해, 원래 데이터를 분할할 때 따로 분리해 두었던 유효성 검사 데이터 세트를 '채점'합니다. 즉, 유효성 검사 데이터 세트의 특징에 대해 레이블을 예측합니다. **모델 채점 및 평가** 섹션을 펼치고 **모델 채점** 모듈을 **모델 학습** 모듈 아래에 있는 캔버스로 끌어다 놓습니다. 그런 다음 **모델 학습** 모듈의 출력을 **모델 채점** 모듈의 **학습된 모델**(왼쪽) 입력에 연결하고 **데이터 분할** 모듈의 결과 데이터 세트 2(오른쪽) 출력을 **모델 채점** 모듈의 **데이터 세트**(오른쪽) 입력에 연결합니다.
8. 파이프라인이 다음과 같아야 합니다.



학습 파이프라인 실행

이제 학습 파이프라인을 실행하고 모델을 학습할 준비가 되었습니다.

- 제출 을 선택하고 `mslearn-diabetes-training` 이라는 기존 실험을 사용하여 파이프라인을 실행합니다.
- 실험이 완료될 때까지 기다립니다. 5분 이상 걸릴 수 있습니다.
- 실험이 완료되면 모델 채점 모듈을 선택하고 설정 창에서 출력 + 로그 탭의 점수가 매겨진 데이터 세트 섹션에 있는 데이터 출력에서 시각화 아이콘을 사용하여 결과를 봅니다.
- 오른쪽으로 스크롤하면 Diabetic 열(레이블의 알려진 실제 값이 포함됨) 옆에 예측 레이블 값을 포함하는 **Scored Labels**(점수가 매겨진 레이블)이라는 새 열이 있으며, 여기에는 예측 레이블 값과 0과 1 사이의 확률을 포함하는 **Scored Probabilities**(점수가 매겨진 확률) 열이 포함됩니다. 이는 ‘양성’ 예측의 확률을 나타내므로 0.5보다 큰 확률은 예측 레이블이 1(당뇨)이 되고 0과 0.5 사이의 확률은 예측 레이블이 ‘0’(당뇨 아님)이 됩니다.*
- 모델 채점 결과 시각화* 창을 닫습니다.

모델에서 Diabetic 레이블의 값을 예측하지만 그 예측은 어느 정도 안정적일까요? 이를 평가려면 모델을 평가해야 합니다.

다음 단원: 분류 모델 평가

< 이전

단위 6/10 ▼

다음 >

✓ 100 XP



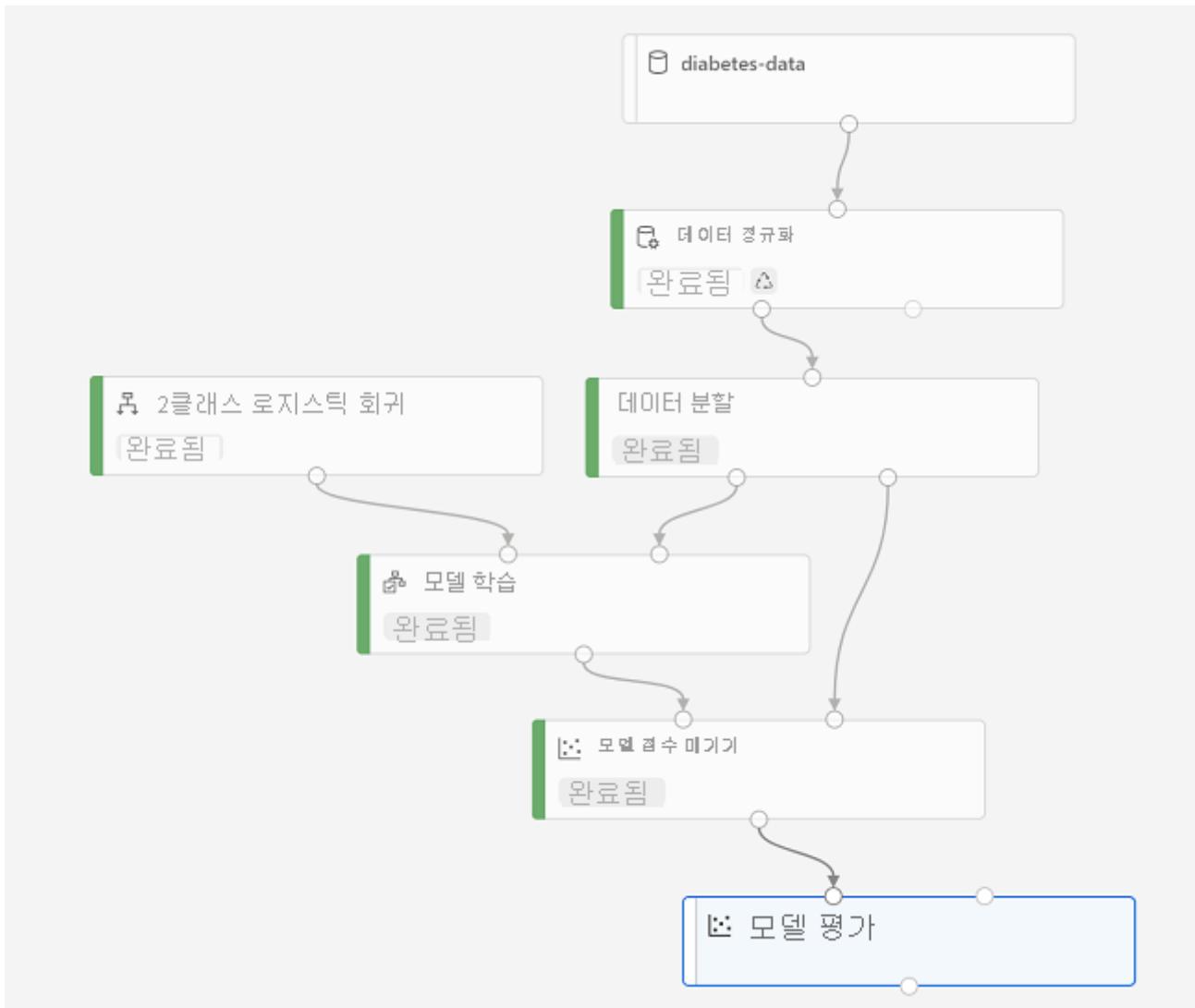
분류 모델 평가

10분

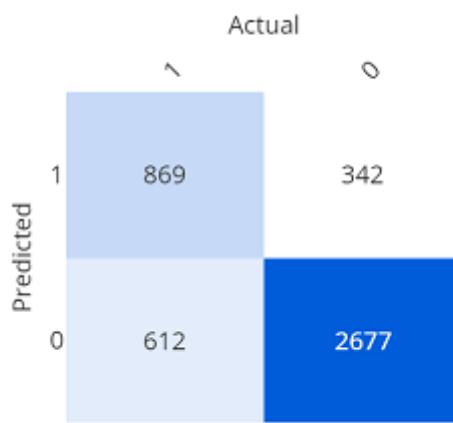
저장해서 모델 채점에 사용한 유효성 검사 데이터에는 레이블에 대해 알려진 값이 포함됩니다. 따라서 모델의 유효성을 검사하기 위해 유효성 검사 데이터 세트를 채점할 때 예측된 레이블 값과 레이블의 true 값을 비교할 수 있습니다. 이 비교에 따라 모델의 성과를 설명하는 다양한 메트릭을 계산할 수 있습니다.

모델 평가 모듈 추가

1. 아직 열려 있지 않다면 이전 단원에서 만든 **당뇨병 학습** 파이프라인을 엽니다.
2. 왼쪽 창의 **모델 채점 및 평가** 섹션에서 **모델 평가** 모듈을 **모델 채점** 모듈 아래에 있는 캔버스로 끌어다 놓고, **모델 채점** 모듈의 출력을 **모델 평가** 모듈의 점수가 매겨진 데이터 세트(왼쪽) 입력에 연결합니다.
3. 파이프라인이 다음과 같아야 합니다.



4. 제출 을 선택하고 mslearn-diabetes-training 이라는 기준 실험을 사용하여 파이프라인을 실행합니다.
5. 실험이 완료될 때까지 기다립니다.
6. 실험이 완료되면 모델 평가 모듈을 선택하고 설정 창에서 출력 + 로그 탭의 평가 결과 섹션에 있는 데이터 출력에서 시각화 아이콘을 사용하여 성능 메트릭을 봅니다. 이러한 메트릭은 데이터 과학자가 유효성 검사 데이터를 기준으로 모델이 얼마나 잘 예측하는지를 평가하는 데 도움이 될 수 있습니다.
7. 가능한 각 클래스에 대해 예측 값과 실제 값 수의 집계인 모델의 혼동 행렬을 확인합니다. 이와 같이 두 가지 가능한 값 중 하나를 예측하는 이진 분류 모델의 경우 혼동 행렬은 클래스 0 과 1에 대한 예측 값과 실제 값을 보여 주는 2x2 표이며, 다음과 유사합니다.



혼동 행렬에는 예측 값과 실제 값이 모두 1(진양성이라고 함)인 경우가 왼쪽 상단에 표시되고 예측 값과 실제 값이 모두 0(진음성)인 사례가 오른쪽 하단에 표시됩니다. 다른 셀에는 예측 값과 실제 값이 서로 다른 경우(위양성 및 위음성)가 표시됩니다. 행렬의 셀에는 색이 지정되어 있으며 셀에 표시되는 사례가 많을 수록 색이 더 진해집니다. 결과에서는 왼쪽 위에서 오른쪽 아래로 색이 진한 셀을 대각선으로 찾아 보면서 모든 클래스에 대해 정확하게 예측하는 모델을 식별할 수 있습니다(즉, 예측 값이 실제 값과 일치하는 셀). 다중 클래스 분류 모델의 경우(가능한 클래스가 두 개 이상인 경우), 동일한 접근 방식이 가능한 각 실제 값 및 예측 값 수의 조합을 테이블화하는 데 사용됩니다. 따라서, 세 가지 클래스가 가능한 모델은 예측 및 실제 레이블 일치 셀이 대각선을 이루는 3x3 행렬이 만들어집니다.

8. 다음이 포함된 혼동 행렬의 왼쪽에서 메트릭을 검토합니다.

- **정확도:** 올바른 예측(진양성 + 진음성)과 총 예측 수의 비율입니다. 즉, 모델이 당뇨 병을 올바르게 예측하는 비율은 어느 정도입니까?
- **정밀도:** 올바르게 식별된 양성 사례의 비율입니다(진양성 수를 진양성 수와 위양성 수의 합으로 나눈 비율). 즉, 모델에서 당뇨병이 있는 것으로 예측한 모든 환자 중에 실제로 당뇨가 있는 환자의 수는 얼마입니까?
- **재현율:** 양성으로 분류된 사례 중 실제로 양성인 비율입니다(진양성 수를 진양성 수와 위음성 수의 합으로 나눈 비율). 즉, 실제로는 당뇨가 있는 모든 환자 중 모델이 식별할 수 있는 환자의 수는 얼마입니까?
- **F1 점수:** 전체 메트릭은 기본적으로 정밀도와 재현율을 결합합니다.

- 나중에 AUC로 돌아갑니다.

이러한 메트릭의 정확도는 가장 직관적입니다. 하지만, 모델의 작동 성과를 측정하는 것만 큼 단순 정확도를 사용하는 데도 주의를 기울여야 합니다. 모집단의 3%만이 당뇨병이 있다고 가정합니다. 항상 0을 예측하는 모델을 만들 수 있으며 이때의 정확도는 97%이지만 아주 유용하지는 않습니다. 이러한 이유로 대부분의 데이터 과학자는 정밀도와 재현율 같은 다른 메트릭을 사용하여 분류 모델의 성능을 평가합니다.

9. 메트릭 목록 위에 **임계값** 슬라이더가 있는지 확인합니다. 분류 모델에서 예측하는 것은 가능한 각 클래스의 확률입니다. 이 이진 분류 모델의 경우 양성(당뇨병 있음)으로 예측될 확률은 0과 1 사이에 있는 값입니다. 기본적으로 당뇨병 예측 확률이 0.5를 넘으면 클래스 예측은 1이 되고, 예측이 임계값보다 작으면 환자가 당뇨병이 **아닐** 확률이 더 크므로(클래스의 확률은 다 더하면 1이 됨) 예측 클래스는 0이 됩니다. 임계값 슬라이더를 이동하여 혼동 행렬에 미치는 영향을 확인하세요. 왼쪽(0)으로 이동하면 재현율 메트릭은 1이 되며, 오른쪽(1)으로 이동하면 재현율 메트릭이 0이 됩니다.
10. ROC 곡선(ROC는 수신된 연산자 특징(Received Operator Characteristic)을 의미하지만 대부분의 데이터 과학자는 ROC 곡선이라고 부름)의 임계값 슬라이드 위를 확인합니다. 재현율에 대한 또 다른 용어는 **진양성 비율**이며, 여기에는 실제 음성 사례 수 대비 양성으로 잘못 식별된 음성 사례의 수를 측정하는 **위양성 비율**이라는 메트릭이 따라옵니다. 이러한 메트릭을 0과 1 사이에서 가능한 모든 임계값에 대해 서로를 대조하면 곡선이 도출됩니다. 이상적인 모델에서는 곡선은 좌상향하는 방향으로 나가 차트의 전체 영역을 포함합니다. 곡선 아래 영역(0~1 범위의 값)이 클수록 모델의 성능이 뛰어나며, 이는 아래에 다른 메트릭과 함께 나열된 **AUC** 메트릭입니다. 이 영역이 모델의 성능을 어떻게 표시하는지 이해하려면 ROC 차트의 왼쪽 아래에서 오른쪽 위로 향하는 대각선을 생각해 보세요. 각 환자의 상태를 추측하거나 동전 던지기로 정하는 경우에 예상되는 결과를 나타냅니다. 반은 맞고 반은 틀릴 거라 예측할 수 있으므로 대각선 아래의 영역은 AUC 0.5를 나타냅니다. 모델에 대한 AUC가 이진 분류 모델의 AUC보다 높을 경우 모델은 임의의 추측보다 나은 결과를 도출합니다.

11. 모델 평가 결과 시각화 창을 닫습니다.

기능 엔지니어링과 전처리를 최소한으로 했기 때문에 이 모델의 성능이 모두 좋은 것은 아닙니다. 2 클래스 의사 결정 포리스트 와 같은 다른 분류 알고리즘을 사용하고 결과를 비교해 볼 수 있습니다. 데이터 분할 모듈의 출력을 여러 학습 모델과 채점 모델 모듈에 연결할 수 있으며 두 번째 채점 모델 모듈을 평가 모델 모듈에 연결하면 나란히 비교할 수도 있습니다. 이 연습에서는 완벽한 모델을 학습하는 것이 아니라 분류 및 Azure Machine Learning 디자이너 인터페이스를 소개하는 것으로 충분합니다.

다음 단원: 유추 파이프라인 만들기

계속 >

< 이전

단위 7/10 ▼

다음 >

✓ 100 XP



유추 파이프라인 만들기

8분

모델을 학습하기 위해 파이프라인을 만들고 실행한 후에는 새 데이터에 대해 동일한 데이터 변환을 수행한 다음 학습된 모델을 사용하여 해당 특징에 따라 레이블 값을 유추(즉, 예측)하는 두 번째 파이프라인이 필요합니다. 이 파이프라인을 통해 사용할 애플리케이션에 게시할 수 있는 예측 서비스의 기반이 형성됩니다.

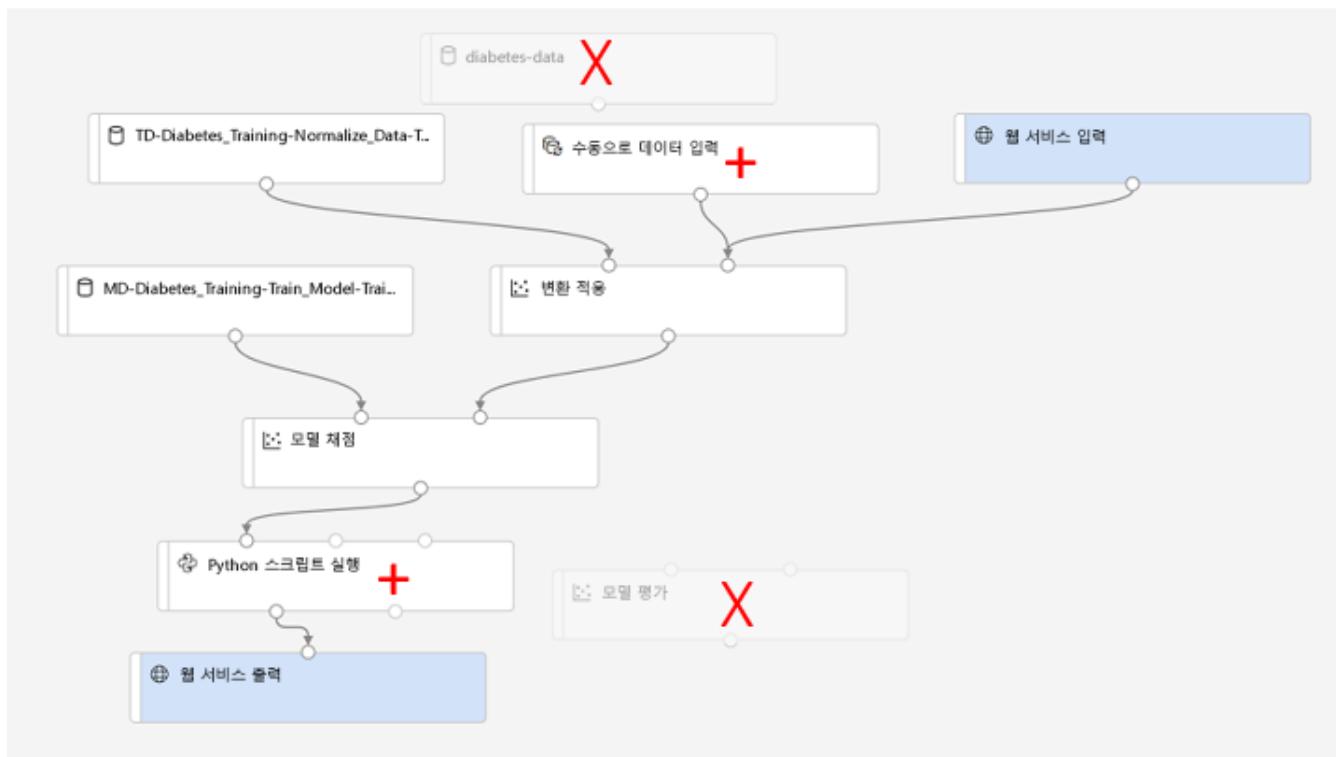
유추 파이프라인 만들기

1. Azure Machine Learning Studio에서 **디자이너** 페이지를 클릭하여 만든 모든 파이프라인을 표시합니다. 그런 다음 이전에 만든 **당뇨병 학습** 파이프라인을 엽니다.
2. **유추 파이프라인 만들기** 드롭다운 목록에서 **실시간 유추 파이프라인**을 클릭합니다. 몇 초 후에 **당뇨병 학습-실시간 유추**라는 새 버전의 파이프라인이 열립니다.

파이프라인에 웹 서비스 입력과 웹 서비스 출력 모듈이 포함되지 않은 경우 **디자이너** 페이지로 돌아가 **당뇨병 학습-실시간 유추** 파이프라인을 다시 엽니다.

3. 새 파이프라인의 이름을 **당뇨병 예측**으로 바꾸고 새 파이프라인을 검토합니다. 제출할 새 데이터에 대한 웹 서비스 입력이 포함되어 있고, 웹 서비스 출력이 포함되어 결과를 반환합니다. 학습 데이터의 통계가 새 데이터 값을 정규화하는 데 사용되고 학습된 모델이 새 데이터의 점수를 매기는 데 사용될 수 있도록 일부 변환 및 학습 단계가 해당 파이프라인에 캡슐화되었습니다.

유추 파이프라인을 다음과 같이 변경합니다.



- **diabetes-data** 데이터 세트를 레이블 열(Diabetic)을 포함하지 않은 **수동으로 데이터 입력** 모듈로 바꿉니다.
- **모델 평가** 모듈을 제거합니다.
- 환자 ID, 예측된 레이블 값, 확률만 반환하도록 웹 서비스 출력 앞에 **Python 스크립트 실행** 모듈을 삽입합니다.

파이프라인을 수정할 때 아래의 단계를 따르고 위 이미지와 정보를 참조합니다.

4. 유추 파이프라인은 새 데이터가 원래 학습 데이터의 스키마와 일치하는 것으로 가정하므로 학습 파이프라인의 **diabetes-data** 데이터 세트가 포함됩니다. 해당 입력 데이터에는 모델이 예측하는 **Diabetic** 레이블이 포함되어 있지만, 아직 당뇨병 예측이 이루어지지 않은 새 환자 데이터에 이를 포함시키는 것은 비논리적입니다. 해당 모듈을 삭제하고 **데이터 입력 및 출력** 섹션에서 **수동으로 데이터 입력** 모듈을 사용하여 세 명의 신규 환자 관찰에 대해 레이블을 제외한 특징 값이 포함된 다음 CSV 데이터가 포함되도록 바꿉니다(텍스트 블록 전체 복사 및 붙여넣기).

CSV	
PatientID,Pregnancies,PlasmaGlucose,DiastolicBloodPressure,TricepsThickness,SerumInsulin,BMI,DiabetesPedigree,Age 1882185,9,104,51,7,24,27.36983156,1.350472047,43 1662484,6,73,61,35,24,18.74367404,1.074147566,75 1228510,4,115,50,29,243,34.69215364,0.741159926,59	

5. 새 **수동으로 데이터 입력** 모듈을 **변환 적용** 모듈의 동일한 **데이터 세트** 입력에 **웹 서비스 입력**으로 연결합니다.

6. 유추 파이프라인에는 새 데이터를 예측할 때 유용하지 않은 **모델 평가** 모듈이 포함되어 있으므로 해당 모듈을 삭제합니다.
7. **모델 채점** 모듈의 출력에는 모든 입력 특징뿐만 아니라 예측된 레이블과 확률 점수도 포함됩니다. 출력을 예측과 확률로만 제한하려면 다음을 수행합니다.

- **모델 채점** 모듈과 웹 서비스 출력 간의 연결을 삭제합니다.
- Python 언어 섹션에서 **Python 스크립트 실행** 모듈을 추가하고 모든 기본 Python 스크립트를 다음 코드로 바꿉니다(PatientID, Scored Labels(점수가 매겨진 레이블), Scored Probabilities(점수가 매겨진 확률) 열만 선택하고 적절하게 이름 변경).

```
Python 복사

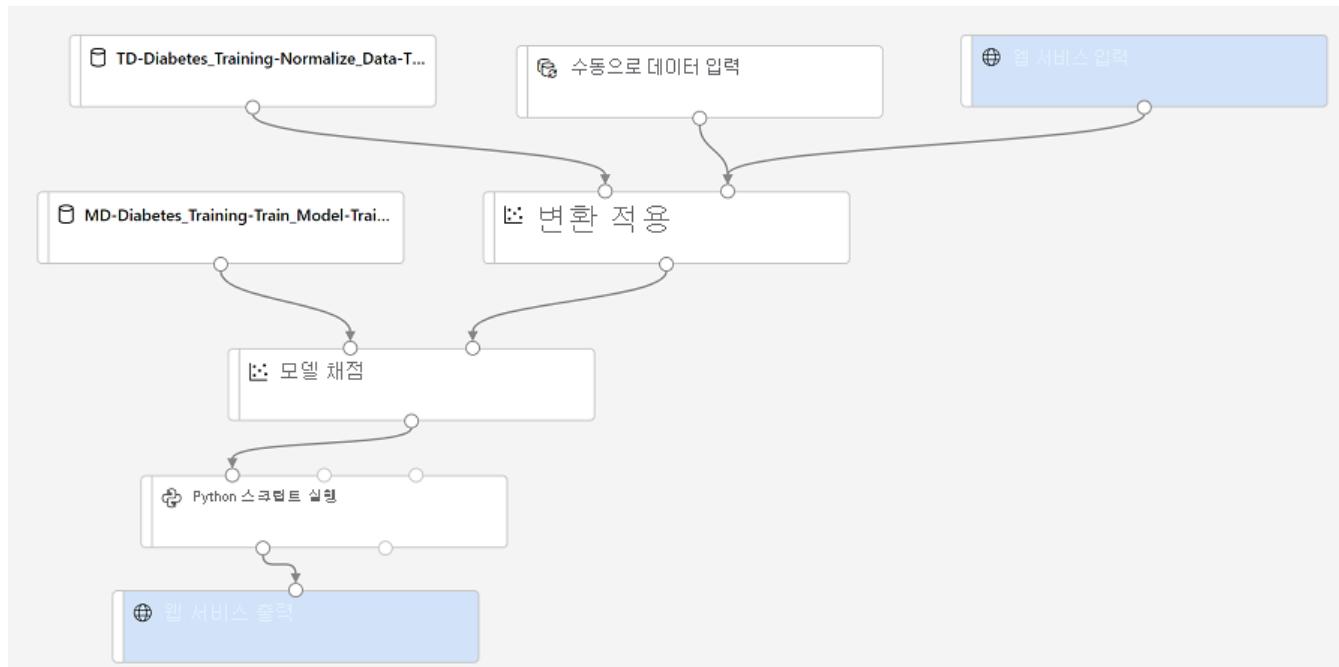
import pandas as pd

def azureml_main(dataframe1 = None, dataframe2 = None):

    scored_results = dataframe1[['PatientID', 'Scored Labels', 'Scored
Probabilities']]
    scored_results.rename(columns={'Scored Labels':'DiabetesPrediction',
                                   'Scored Probabilities':'Probability'},
                           inplace=True)
    return scored_results
```

- 모델 채점 모듈의 출력을 **Python 스크립트 실행** 의 Dataset1(맨 왼쪽) 입력에 연결하고 **Python 스크립트 실행** 모듈의 출력을 웹 서비스 출력에 연결합니다.

8. 파이프라인이 다음과 유사한지 확인합니다.



9. 컴퓨팅 클러스터에서 **mslearn-diabetes-inference**라는 새 실험으로 파이프라인을 실행합니다. 다소 시간이 걸릴 수 있습니다.
10. 파이프라인이 완료되면 **Python 스크립트 실행** 모듈을 선택하고, 설정 창의 **출력 + 로그** 탭에서 **결과 데이터 세트**를 시각화하여 입력 데이터의 세 환자 관찰값에 대한 예측 레이블과 확률을 확인합니다.

유추 파이프라인은 특징을 기준으로 환자에게 당뇨병 위험이 있는지 여부를 예측합니다. 이제 클라이언트 애플리케이션에서 사용할 수 있도록 파이프라인을 게시할 준비가 되었습니다.

다음 단원: 예측 서비스 배포

계속 >

< 이전

단위 8/10 ▼

다음 >

✓ 100 XP



예측 서비스 배포

5분

실시간 추론용 추론 파이프라인을 만들고 테스트한 후에는 이를 클라이언트 응용 프로그램에서 사용할 서비스로 게시할 수 있습니다.

① 참고

이 연습에서는 ACI(Azure Container Instance)에 웹 서비스를 배포합니다. 이러한 유형의 컴퓨팅은 동적으로 만들어지며 개발 및 테스트에 유용합니다. 프로덕션 환경에서는 '유추 클러스터'를 만들어 향상된 스케일링 성능 및 보안을 제공하는 AKS(Azure Kubernetes Service) 클러스터를 제공해야 합니다.

서비스 배포

1. 이전 단원에서 만든 **당뇨병 예측** 유추 파이프라인을 봅니다.
2. 오른쪽 위에서 **배포**를 선택하고 다음 설정을 사용하여 새로운 실시간 엔드포인트를 배포합니다.
 - **이름:** predict-diabetes
 - **설명:** 당뇨병을 분류합니다.
 - **컴퓨팅 형식:** Azure Container Instances
3. 웹 서비스가 배포될 때까지 기다립니다. 몇 분 정도 걸릴 수 있습니다. 배포 상태는 디자이너 인터페이스의 왼쪽 상단에 표시됩니다.

서비스 테스트

이제 클라이언트 애플리케이션에서 배포된 서비스를 테스트할 수 있습니다. 이 경우 아래 셀의 코드를 사용하여 클라이언트 애플리케이션을 시뮬레이션합니다.

1. **엔드포인트** 페이지에서 **predict-diabetes** 실시간 엔드포인트를 엽니다.
2. **predict-diabetes** 엔드포인트가 열리면 **사용** 탭을 보고 다음 정보를 확인합니다. 클라이언트 애플리케이션에서 배포된 서비스를 연결하려면 해당 정보가 필요합니다.
 - 서비스에 대한 REST 엔드포인트
 - 서비스에 대한 기본 키

3. 이 값 옆에 있는 링크를 사용하여 이를 클립보드에 복사할 수 있습니다.
4. 브라우저에서 **predict-diabetes** 서비스 페이지의 사용 페이지가 열린 상태에서 새 브라우저 탭을 열고 **Azure Machine Learning Studio** 의 두 번째 인스턴스를 엽니다. 그런 다음 새 탭에서 **Notebooks** 페이지(작성자 아래)를 봅니다.
5. **Notebooks** 페이지의 **내 파일**에서 단추를 사용하여 다음 설정으로 새 파일을 만듭니다.
 - **파일 위치:** Users/*your user name*
 - **파일 이름:** Test-Diabetes
 - **파일 형식:** Notebook
 - **이미 있는 경우 덮어쓰기:** 선택됨
6. 새 Notebook이 만들어지면 이전에 만든 컴퓨팅 인스턴스가 **컴퓨팅** 상자에서 선택되었는지, 그리고 상태가 **실행 중**인지 확인합니다.
7. << 단추를 사용하여 파일 탐색기 창을 축소하고 **Test-Diabetes.ipynb** Notebook 탭에 사용 할 수 있는 공간을 제공합니다.
8. Notebook에서 만든 사각형 셀에 다음 코드를 붙여넣습니다.

Python

복사

```
endpoint = 'YOUR_ENDPOINT' #Replace with your endpoint
key = 'YOUR_KEY' #Replace with your key

import urllib.request
import json
import os

data = {
    "Inputs": {
        "WebServiceInput0": [
            {
                "PatientID": 1882185,
                "Pregnancies": 9,
                "PlasmaGlucose": 104,
                "DiastolicBloodPressure": 51,
                "TricepsThickness": 7,
                "SerumInsulin": 24,
                "BMI": 27.36983156,
                "DiabetesPedigree": 1.3504720469999998,
                "Age": 43,
            },
        ],
    },
    "GlobalParameters": {
    }
}
```

```

body = str.encode(json.dumps(data))

headers = { 'Content-Type':'application/json', 'Authorization':('Bearer ' +
key)}

req = urllib.request.Request(endpoint, body, headers)

try:
    response = urllib.request.urlopen(req)
    result = response.read()
    json_result = json.loads(result)
    output = json_result["Results"][ "WebServiceOutput0"] [0]
    print('Patient: {} \nPrediction: {} \nProbability:
{:.2f}'.format(output[ "PatientID"],
output[ "DiabetesPrediction"],
output[ "Probability"]))
except urllib.error.HTTPError as error:
    print("The request failed with status code: " + str(error.code))

    # Print the headers to help debug
    print(error.info())
    print(json.loads(error.read().decode("utf8", 'ignore')))
```

① 참고

코드의 세부 정보에 대해 너무 걱정하지 마세요. 환자의 특징을 정의하고 당뇨병 진단 예측을 위해 만든 predict-diabetes 서비스를 사용합니다.

9. predict-diabetes 서비스의 **사용** 페이지가 포함된 브라우저 탭으로 전환한 다음 서비스에 대한 REST 엔드포인트를 복사합니다. Notebook을 포함하는 탭으로 돌아가 키를 코드에 복사하고 YOUR_ENDPOINT를 대체합니다.
10. predict-diabetes 서비스의 **사용** 페이지가 포함된 브라우저 탭으로 전환한 다음 서비스에 대한 기본 키를 복사합니다. Notebook을 포함하는 탭으로 돌아가 키를 코드에 복사하고 YOUR_KEY를 대체합니다.
11. Notebook을 저장합니다. 셀 옆에 있는 ▶ 단추를 사용하여 코드를 실행합니다.
12. 예측된 당뇨병 진단이 반환되는지 확인합니다.

다음 단원: 지식 점검

계속 >



✓ 1100 XP ➔

Azure Machine Learning 디자이너를 사용하여 클러스터링 모델 만들기

49분 • 모듈 • 10 단위

★★★★★ 4.8 (1,601) [평가하기](#)

초급 AI 엔지니어 데이터 과학자 Machine Learning

클러스터링은 특징을 기반으로 유사한 엔터티를 그룹화하는 데 사용되는 자율 기계 학습 기술입니다. Azure Machine Learning 디자이너를 사용하여 클러스터링 모델을 만드는 방법을 알아봅니다.

학습 목표

Azure Machine Learning 디자이너를 사용하여 클러스터링 모델 학습 및 게시

[책갈피](#) [컬렉션에 추가](#)

필수 조건

Azure Portal 탐색 능력

이 모듈은 이러한 학습 경로의 일부입니다.[Azure Machine Learning을 사용하여 코드 없는 예측 모델 만들기](#)

소개

2분



Azure Machine Learning 작업 영역 만들기

5분



컴퓨팅 리소스 만들기

5분



데이터 탐색

8분



학습 파이프라인 만들기 및 실행

8분



클러스터링 모델 평가

5분



유추 파이프라인 만들기



8분

예측 서비스 배포

5분

지식 점검

2분

요약

1분

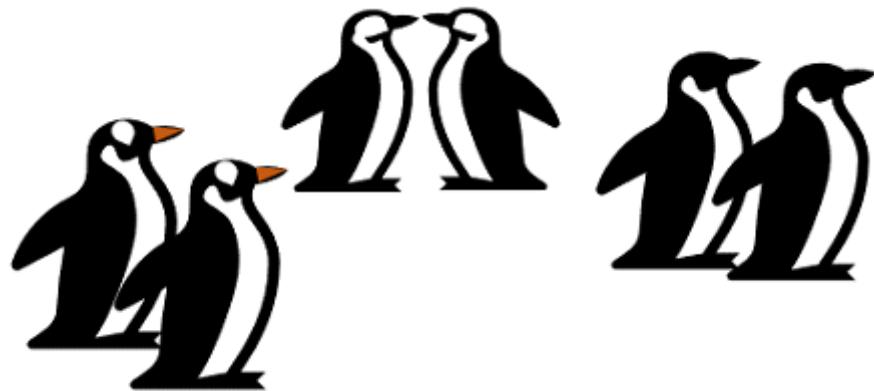
✓ 100 XP



소개

2분

클러스터링은 유사한 항목을 해당 특징에 따라 클러스터로 그룹화하는 데 사용되는 기계 학습의 한 형태입니다. 예를 들어 연구원은 펭귄을 측정하고 비율의 유사성에 따라 그룹화할 수 있습니다.



클러스터링은 감독되지 않은 기계 학습의 한 예로, 해당 특성 또는 기능에 따라 항목을 클러스터로 분리하는 모델을 학습합니다. 모델을 학습하는 이전에 알려진 클러스터 값(또는 레이블)은 없습니다.

Microsoft Azure Machine Learning 디자이너를 사용하여 코드를 작성할 필요 없이 끌어다 놓기 시각적 인터페이스를 사용하여 클러스터링 모델을 만들 수 있습니다.

이 모듈에서는 다음 방법을 알아봅니다.

- Azure Machine Learning 디자이너를 사용하여 클러스터링 모델을 학습합니다.
- 추론을 위해 클러스터링 모델을 사용합니다.
- 클러스터링 모델을 서비스로 배포합니다.

이 모듈을 완료하려면 Microsoft Azure 구독이 필요합니다. 구독이 아직 없다면 <https://azure.microsoft.com>에서 평가판에 가입할 수 있습니다.

다음 단원: Azure Machine Learning 작업 영역 만들기

계속 >

< 이전

단위 2/10 ▼

다음 >

✓ 100 XP



Azure Machine Learning 작업 영역 만들기

5분

Azure Machine Learning은 Azure에서 기계 학습 솔루션을 빌드하고 운영하기 위한 클라우드 기반 플랫폼입니다. 데이터 과학자가 데이터를 준비하고, 모델을 학습하고, 예측 서비스를 게시하고, 사용량을 모니터링하는 데 도움이 되는 여러 특징 및 기능을 포함합니다. 해당 특징 중 하나는 '디자이너'라는 시각적 인터페이스로, 코드 작성 없이 기계 학습 모델을 학습, 테스트 및 배포할 수 있게 해 줍니다.

Azure Machine Learning 작업 영역 만들기

Azure Machine Learning을 사용하려면 Azure 구독에서 '작업 영역'을 만듭니다. 그런 다음 이 작업 영역을 사용하여 데이터, 컴퓨팅 리소스, 코드, 모델 및 기계 학습 워크로드와 관련된 기타 아티팩트를 관리할 수 있습니다.

ⓘ 참고

이 모듈은 Azure Machine Learning 작업 영역을 사용하는 여러 가지 모듈 중 하나입니다.

Azure AI 기본 사항 또는 **Azure 데이터 과학자** 인증을 준비하는 과정에서 이 모듈을 진행하는 경우, 작업 영역을 한 번 만들어 두고 다른 모듈에서 재사용하는 것이 좋습니다. 각 모듈을 완료한 후에는 모듈 끝에 있는 정리 지침에 따라 컴퓨팅 리소스를 중지해야 합니다.

아직 없는 경우 다음 단계에 따라 작업 영역을 만듭니다.

1. Microsoft 자격 증명을 사용하여 [Azure Portal](#) 에 로그인합니다.
2. + 리소스 만들기를 선택하고, 'Machine Learning'을 검색하고, 다음 설정을 포함하여 새 Machine Learning 리소스를 만듭니다.

- **구독:** 'Azure 구독'
- **리소스 그룹:** '리소스 그룹 만들기 또는 선택'
- **작업 영역 이름:** '작업 영역의 고유한 이름 입력'
- **지역:** '지리적으로 가장 가까운 지역 선택'
- **스토리지 계정:** '작업 영역에 대해 만들 새로운 기본 스토리지 계정'
- **키 자격 증명 모음:** '작업 영역에 대해 만들 새로운 기본 키 자격 증명 모음'
- **Application insights:** '작업 영역에 대해 만들 새로운 기본 Application Insights 리소스'
- **컨테이너 레지스트리:** 없음('처음으로 컨테이너에 모델을 배포할 때 자동으로 만들 어짐')

3. 작업 영역이 만들어질 때까지 기다립니다(몇 분이 걸릴 수 있음). 그런 다음 포털에서 해당 위치로 이동합니다.
4. 작업 영역의 **개요** 페이지에서 Azure Machine Learning Studio를 실행(또는 새 브라우저 탭을 열고 <https://ml.azure.com> 으로 이동)한 다음 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다.
5. Azure Machine Learning Studio에서 왼쪽 상단에 있는 ☰ 아이콘을 설정/해제하면 인터페이스에 다양한 페이지가 표시됩니다. 해당 페이지를 사용하여 작업 영역에서 리소스를 관리할 수 있습니다.

Azure Portal을 사용하여 작업 영역을 관리할 수 있지만 데이터 과학자 및 기계 학습 운영 엔지니어의 경우 Azure Machine Learning Studio에서 작업 영역 리소스를 관리하는 데 특화된 사용자 인터페이스를 이용할 수 있습니다.

다음 단원: 컴퓨팅 리소스 만들기

계속 >

< 이전

단위 3/10 ▼

다음 >

✓ 100 XP



컴퓨팅 리소스 만들기

5분

Azure Machine Learning 디자이너를 사용하여 모델을 학습하고 배포하려면 학습 프로세스를 실행하고, 배포 후 학습된 모델을 테스트하기 위한 컴퓨팅이 필요합니다.

컴퓨팅 대상 만들기

컴퓨팅 대상은 모델 학습 및 데이터 탐색 프로세스를 실행할 수 있는 클라우드 기반 리소스입니다.

1. [Azure Machine Learning Studio](#)에서 관리 아래에 있는 컴퓨팅 페이지를 확인합니다. 해당 페이지는 데이터 과학 관련 활동을 위해 컴퓨팅 대상을 관리하는 곳입니다. 다음 네 가지 종류의 컴퓨팅 리소스를 만들 수 있습니다.

- **컴퓨팅 인스턴스**: 데이터 과학자가 데이터 및 모델을 작업하는 데 사용할 수 있는 개발 워크스테이션입니다.
- **컴퓨팅 클러스터**: 실험 코드의 주문형 처리를 지원하는 확장 가능한 가상 클러스터입니다.
- **유추 클러스터**: 학습된 모델을 사용하는 예측 서비스의 배포 대상입니다.
- **연결된 컴퓨팅**: Virtual Machines 또는 Azure Databricks 클러스터와 같은 기존 Azure 컴퓨팅 리소스에 연결합니다.

2. 컴퓨팅 인스턴스 탭에서 다음 설정을 사용하여 새 컴퓨팅 인스턴스를 추가합니다. 이 인스턴스를 모델을 테스트할 워크스테이션으로 사용합니다.

- **가상 머신 유형**: CPU
- **가상 머신 크기**: Standard_DS11_v2(모든 옵션에서 선택을 선택하여 이 가상 머신 크기를 검색하여 선택함)
- **컴퓨팅 이름**: 고유한 이름 입력
- **SSH 액세스 사용**: 선택 안 함

3. 컴퓨팅 인스턴스가 생성되는 동안 컴퓨팅 클러스터 탭으로 전환하고 다음 설정을 사용하여 새 컴퓨팅 클러스터를 추가합니다. 이를 사용하여 기계 학습 모델을 학습합니다.

- **가상 머신 우선 순위**: 전용
- **가상 머신 유형**: CPU
- **가상 머신 크기**: Standard_DS11_v2(모든 옵션에서 선택을 선택하여 이 가상 머신 크기를 검색하여 선택함)

- **컴퓨팅 이름:** 고유한 이름 입력
- **최소 노드 수:** 0
- **최대 노드 수:** 2
- **스케일 다운 전 유휴 시간(초):** 120
- **SSH 액세스 사용:** 선택 안 함

① 참고

이 모듈을 완료하지 않기로 결정한 경우 컴퓨팅 인스턴스를 중지하여 Azure 구독에 불필요한 요금이 발생되지 않도록 해야 합니다.

컴퓨팅 대상은 만들어지는 데 시간이 걸립니다. 대기하는 동안 다음 단원으로 이동할 수 있습니다.

다음 단원: 데이터 탐색

계속 >

< 이전

단위 4/10 ▼

다음 >

✓ 100 XP



데이터 탐색

8분

클러스터링 모델을 학습하려면 클러스터링할 항목의 다양한 관찰 결과를 포함하는 데이터 세트가 필요합니다. 여기에는 개별 사례를 클러스터로 분리하는 데 유용한 개별 사례 간의 유사성을 확인할 수 있는 숫자 기능이 포함됩니다.

데이터 세트 만들기

Azure Machine Learning에서 모델 학습 및 기타 작업의 데이터는 주로 데이터 세트라는 개체에 캡슐화됩니다. 이 모듈에서는 세 가지 종의 펭귄에 대한 관찰을 포함하는 데이터 세트를 사용합니다.

1. [Azure Machine Learning Studio](#)에서 **데이터 세트** 페이지를 확인합니다. 데이터 세트는 Azure ML에서 사용할 특정 데이터 파일이나 테이블을 나타냅니다.
2. 다음 설정을 사용하여 웹 파일에서 데이터 세트를 만듭니다.

- **기본 정보:**
 - 웹 URL: <https://aka.ms/penguin-data>
 - 이름: penguin-data
 - 데이터 세트 형식: 테이블 형식
 - 설명: 펭귄 데이터
- **설정 및 미리 보기:**
 - 파일 형식: 구분 기호로 분리됨
 - 구분 기호: 쉼표
 - 인코딩: UTF-8
 - 열 헤더: 첫 번째 파일의 헤더 사용
 - 행 건너뛰기: 없음
- **스키마:**
 - 경로 이외의 모든 열 포함
 - 자동으로 검색된 형식 검토
- **세부 정보 확인:**
 - 만든 후 데이터 세트를 프로파일링하지 않음

3. 데이터 세트를 만든 후에는 이를 열고 **탐색** 페이지를 보면서 데이터 샘플을 확인합니다. 이 데이터는 여러 번 펭귄을 관찰하면서 얻은 부리 길이와 두께, 날개 길이, 무게 측정값을 나타냅니다. 데이터 세트에는 세 가지 종의 펭귄이 표시되어 있습니다. 아멜리 펭귄, 젠투 펭귄, 턱끈 펭귄입니다.

① 참고

이 연습에서 사용되는 펭귄 데이터 세트는 Kristen Gorman 박사와 Long Term Ecological Research Network 의 일원인 Palmer Station, Antarctica LTER 에서 수집 및 제공한 것입니다.

파이프라인 만들기

Azure Machine Learning 디자이너를 시작하려면 먼저 파이프라인을 만들고 작업하려는 데이터 세트를 추가해야 합니다.

1. 작업 영역의 [Azure Machine Learning Studio](#)에서 **디자이너** 페이지를 보고 새 파이프라인을 만듭니다.
2. **설정** 창에서 기본 파이프라인 이름(*Pipeline-Created-on- date***)을 **Train Penguin Clustering**으로 변경합니다(설정 창이 표시되지 않은 경우 상단 파이프라인 이름 옆에 있는 ☰ 아이콘을 클릭).
3. 파이프라인을 실행할 컴퓨팅 대상을 지정해야 합니다. 설정 창에서 **컴퓨팅 대상 선택**을 클릭하고 이전에 만든 컴퓨팅 클러스터를 선택합니다.
4. 디자이너의 왼쪽 창에서 **데이터 세트** 섹션을 확장한 다음 이전 연습에서 만든 데이터 세트를 캔버스에 끌어다 놓습니다.
5. 캔버스에 있는 **penguin-data** 데이터 세트를 마우스 오른쪽 단추로 클릭(Mac의 경우 Ctrl+클릭)하고 **시각화** 메뉴에서 **데이터 세트 출력**을 선택합니다.
6. 데이터의 스키마를 검토하여 다양한 열의 분포를 히스토그램으로 확인할 수 있습니다. 그런 다음 **CulmenLength** 열을 선택합니다. 데이터 세트는 다음과 비슷합니다.



7. 데이터 세트의 특성은 다음과 같습니다.

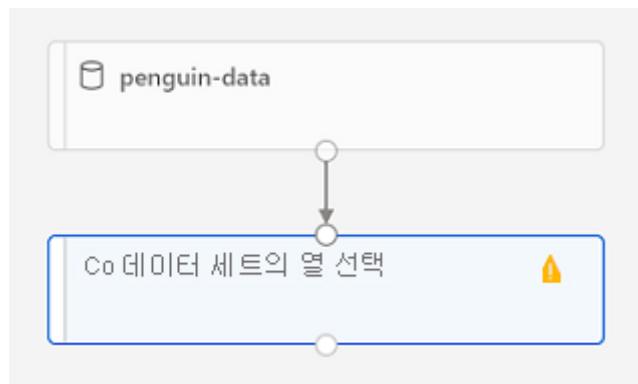
- 데이터 세트에는 다음 열이 포함되어 있습니다.
 - CulmenLength: 펭귄의 부리 길이(밀리미터)입니다.
 - CulmenDepth: 펭귄의 부리 깊이(밀리미터)입니다.
 - FlipperLength: 펭귄의 날개 길이(밀리미터)입니다.
 - BodyMass: 펭귄의 무게(그램)입니다.
 - Species: 종을 나타냅니다(0:"Amelie", 1:"Gentoo", 2:"Chinstrap").
- CulmenLength 열에 누락된 값이 2개 있습니다(CulmenDepth, FlipperLength 및 BodyMass 열에도 누락된 값이 2개 있음).
- 측정값은 규모가 다릅니다(수십 밀리미터에서 수천 그램까지).

8. 파이프라인 캔버스의 데이터 세트를 볼 수 있도록 데이터 세트 시각화를 닫습니다.

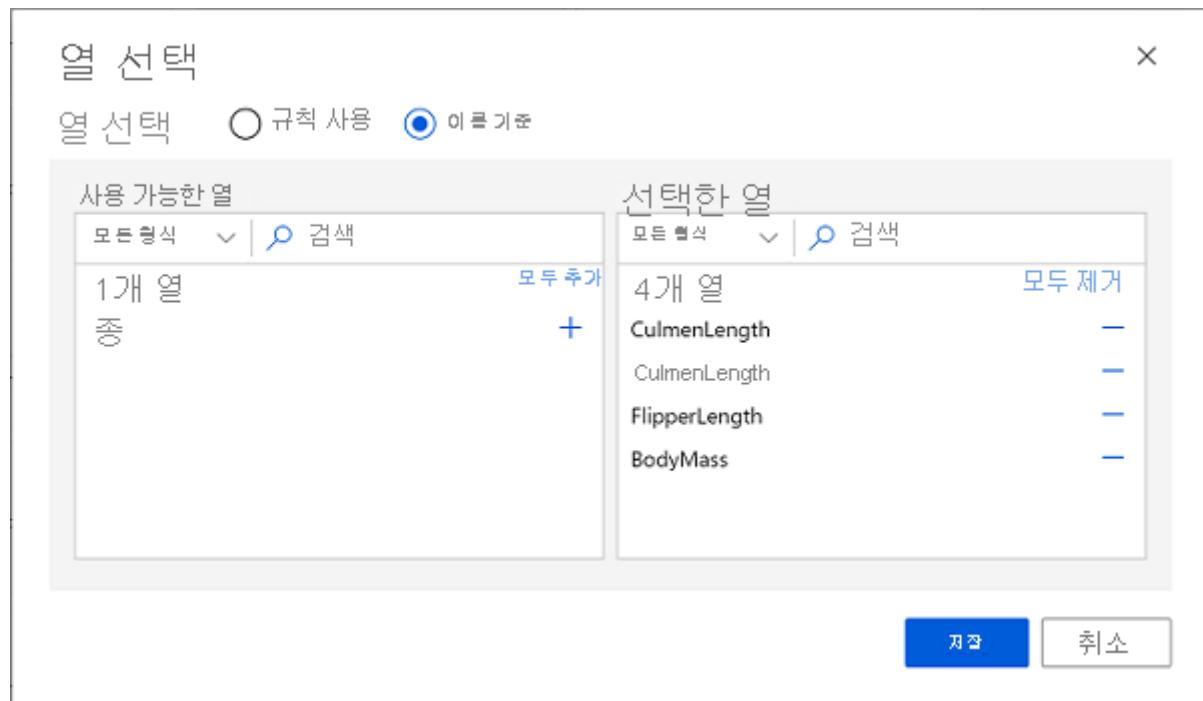
변환 적용

펭귄 관찰 결과를 클러스터링하기 위해 측정값만을 사용하고 종 열은 버리도록 하겠습니다. 또한 값이 누락된 행을 제거하고, 규모가 비슷해지도록 숫자 측정값을 정규화해야 합니다.

1. 왼쪽 창에서 **데이터 변환** 섹션을 펼칩니다. 여기에는 모델 학습 전에 데이터 변환에 사용 할 수 있는 다양한 모듈이 포함되어 있습니다.
2. 펭귄 관찰을 클러스터링하기 위해 측정값만을 사용하고 종 열은 무시하도록 하겠습니다. 따라서 **데이터 세트에서 열 선택** 모듈을 penguin-data 모듈 아래에 있는 캔버스로 끌어다 놓고 penguin-data 모듈 하단의 출력을 **데이터 세트에서 열 선택** 모듈 상단의 입력에 연결합니다.



3. 데이터 세트에서 열 선택 모듈을 선택하고 오른쪽에 있는 설정 창에서 **열 편집** 을 선택합니다. 그런 다음 열 선택 창에서 **이름 기준** 을 선택하고 + 링크를 사용하여 열 이름 CulmenLength, CulmenDepth, FlipperLength 및 BodyMass 를 선택합니다.



4. 데이터 세트에서 열 선택 모듈 설정을 저장하고 디자이너 캔버스로 돌아옵니다.
5. 누락된 데이터 정리 모듈을 데이터 세트에서 열 선택 모듈 아래에 있는 캔버스로 끌어다 놓은 다음 이렇게 연결합니다.



6. 누락된 데이터 정리 모듈을 선택하고 오른쪽의 설정 창에서 열 편집을 클릭합니다. 그런 다음 열 선택 창에서 규칙 사용을 선택하고 모든 열을 포함합니다.



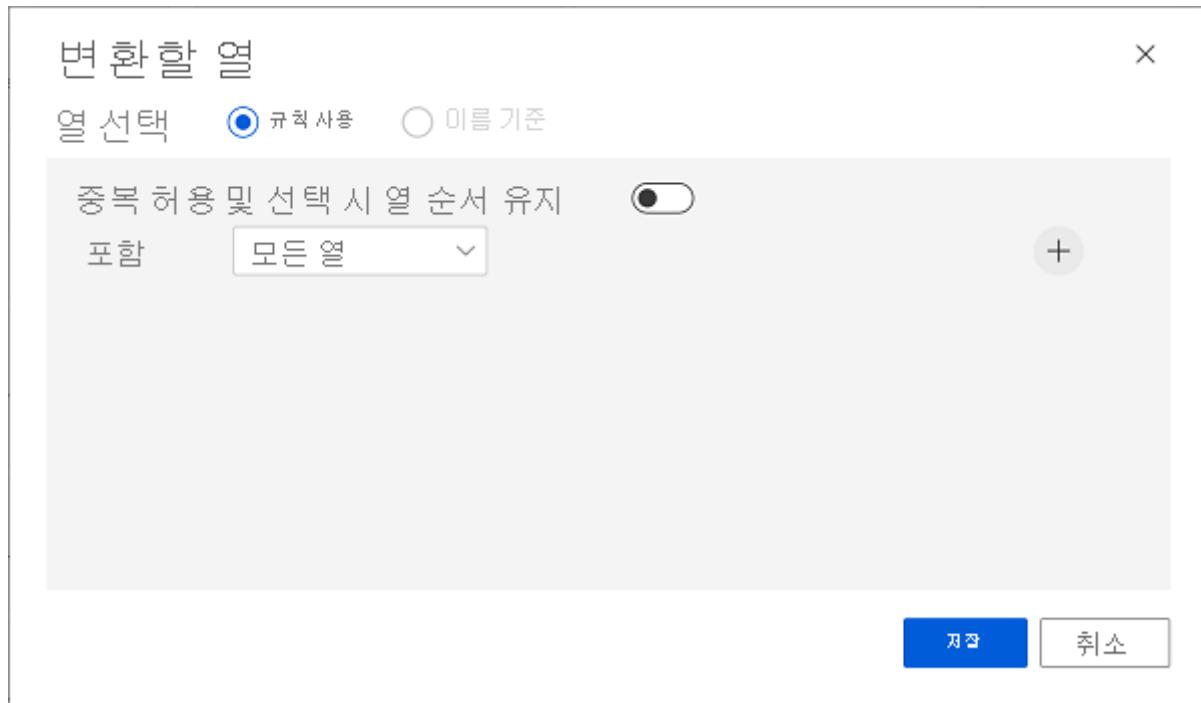
7. 누락된 데이터 정리 모듈을 선택한 상태로 설정 창에서 다음 구성은 설정합니다.

- **최소 누락 값 비율:** 0.0
- **최대 누락 값 비율:** 1.0
- **정리 모드:** 전체 행 제거

8. 데이터 정규화 모듈을 누락된 데이터 정리 모듈 아래에 있는 캔버스로 끌어다 놓습니다. 그런 다음 누락된 데이터 정리 모듈의 가장 왼쪽에 있는 출력과 데이터 정규화 모듈의 입력을 연결합니다.



9. 데이터 정규화 모듈을 선택하고 오른쪽에 있는 설정 창에서 **변환 메서드**를 MinMax로 설정하고 **열 편집**을 선택합니다. 그런 다음 열 선택 창에서 **규칙 사용**을 선택하고 모든 열을 포함합니다.



10. 데이터 정규화 모듈 설정을 저장하고 디자이너 캔버스로 돌아옵니다.

파이프라인 실행

데이터 변환을 적용하려면 파이프라인을 실험으로 실행해야 합니다.

1. 파이프라인이 다음과 유사해야 합니다.



- 제출을 선택하고, 컴퓨팅 클러스터에서 `mslearn-penguin-training`이라는 새 실험을 사용하여 파이프라인을 실행합니다.
- 실행이 끝날 때까지 기다립니다. 5분 이상 걸릴 수 있습니다. 실행이 완료되면 모듈은 다음과 같습니다.



변환된 데이터 보기

이제 모델 학습을 위한 데이터 세트가 준비되었습니다.

1. 완료된 **데이터 정규화** 모듈을 선택하고 오른쪽 **설정** 창의 **출력 + 로그** 탭에서 **변환된 데이터 세트**에 대한 **시각화** 아이콘을 선택합니다.
2. 데이터를 보면 **Species** 열이 제거되어 누락된 값이 없고 4가지 모든 특징에 대한 값이 일 반적인 규모로 정규화되었음을 알 수 있습니다.
3. 정규화된 데이터의 결과 시각화를 닫습니다.

데이터 세트에서 사용하려는 특징을 선택 및 준비했으므로 이제 이를 사용하여 클러스터링 모델을 학습할 준비가 되었습니다.

다음 단원: 학습 파이프라인 만들기 및 실행

계속 >

< 이전

단위 5/10 ▼

다음 >

✓ 100 XP



학습 파이프라인 만들기 및 실행

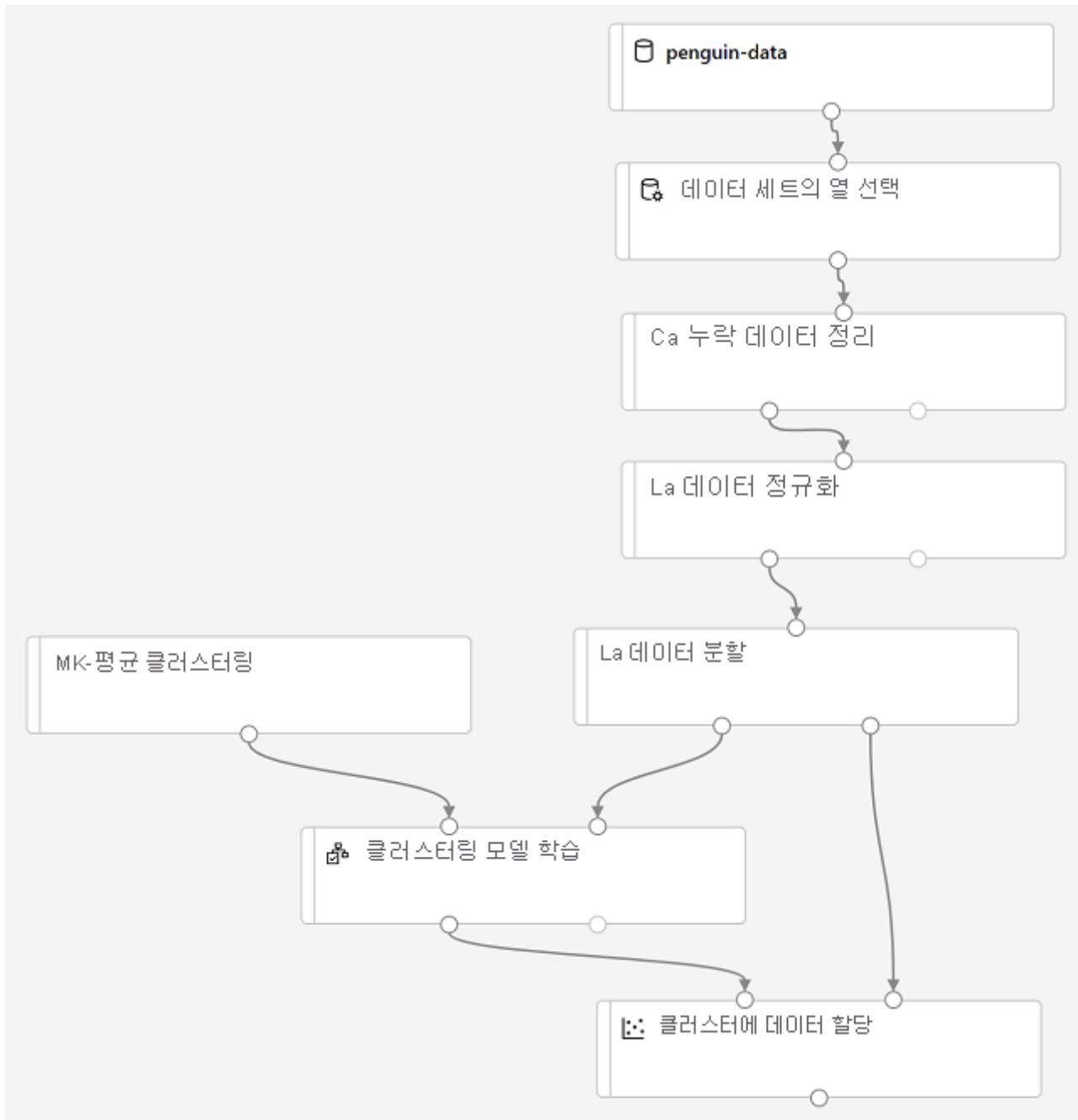
8분

데이터 변환을 사용하여 데이터를 준비한 후에는 이를 사용하여 기계 학습 모델을 학습할 수 있습니다.

학습 모듈 추가

클러스터링 모델을 학습하려면 데이터에 클러스터링 알고리즘을 적용하여 클러스터링을 위해 선택한 특징만을 사용해야 합니다. 데이터의 하위 집합을 사용하여 모델을 학습하고 나머지를 사용하여 학습된 모델을 테스트합니다.

이 연습에서는 다음과 같이 **Train Penguin Clustering** 파이프라인을 확장합니다.

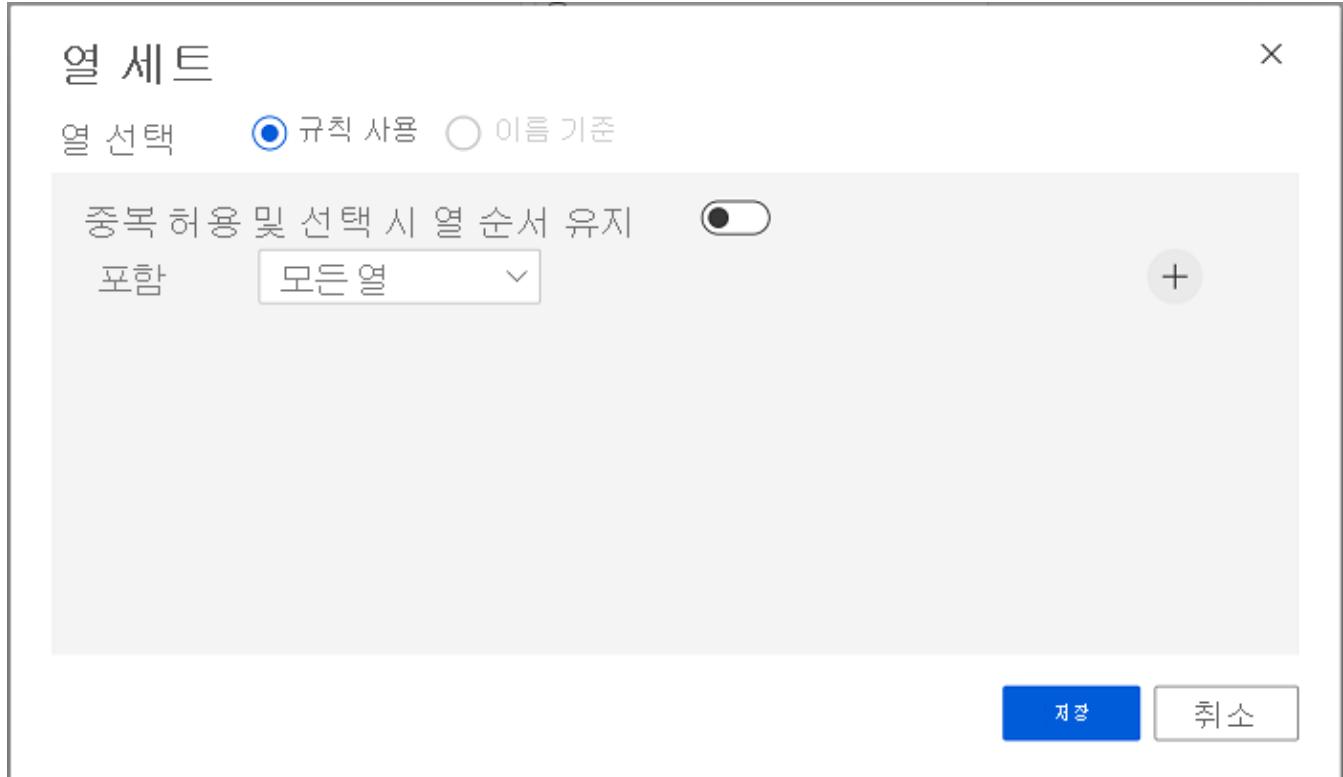


필요한 모듈을 추가하고 구성하는 동안 위 정보를 참조하여 아래의 단계를 따릅니다.

1. 아직 열려 있지 않은 경우 **Train Penguin Clustering** 파이프라인을 엽니다.
2. 왼쪽 창의 **데이터 변환** 섹션에서 **데이터 분할** 모듈을 **데이터 정규화** 모듈의 아래에 있는 캔버스로 끌어 놓습니다. 그런 다음 **데이터 정규화** 모듈의 왼쪽 출력과 **데이터 분할** 모듈의 입력을 연결합니다.
3. **데이터 분할** 모듈을 선택하고, 다음과 같이 설정을 구성합니다.
 - **분할 모드:** 행 분할
 - **첫 번째 출력 데이터 세트에서 행의 비율:** 0.7
 - **무작위 초기값:** 123
 - **계층화된 분할:** 아니요
4. 왼쪽 창에서 **모델 학습** 섹션을 확장하고 **데이터 분할** 모듈에 있는 캔버스로 **클러스터링 모델 학습** 모듈을 끌어다 놓습니다. 그런 다음 **분할 데이터** 모듈의 *Result dataset1(왼쪽)* 출력

력을 **클러스터링 모델 학습** 모듈의 데이터 세트(오른쪽) 입력에 연결합니다.

- 클러스터링 모델은 원래 데이터 세트에서 선택한 모든 특징을 사용하여 데이터 항목에 클러스터를 할당해야 합니다. **클러스터링 모델 학습** 모듈을 선택하고 설정 창에 있는 **매개 변수** 탭에서 **열 편집**을 선택하고 **규칙 사용** 옵션을 사용하여 다음과 같이 모든 열을 포함합니다.



- 학습 중인 모델은 특징을 사용하여 데이터를 클러스터로 그룹화하므로 클러스터링 알고리즘을 사용하여 모델을 학습해야 합니다. **기계 학습 알고리즘** 섹션을 확장한 뒤, **클러스터링**에서 **K-평균 클러스터링** 모듈을 **penguin-data** 데이터 세트의 왼쪽, 그리고 **클러스터링 모델 학습** 모듈 위에 있는 캔버스로 끌어다 놓습니다. 그런 다음 출력을 **클러스터링 모델 학습** 모듈의 **학습되지 않은 모델**(왼쪽) 입력에 연결합니다.

- 'K-평균' 알고리즘은 항목을 지정한 클러스터의 수, 즉 K 값으로 그룹화합니다. ***K-평균 클러스터링****을 선택하고 설정 창에 있는 **매개 변수** 탭에서 **중심의 수** 매개 변수를 3으로 설정합니다.

① 참고

펭귄 측정값과 같은 데이터 관찰을 다차원 벡터라고 생각할 수 있습니다. K-평균은 다음과 같은 방식으로 작동합니다.

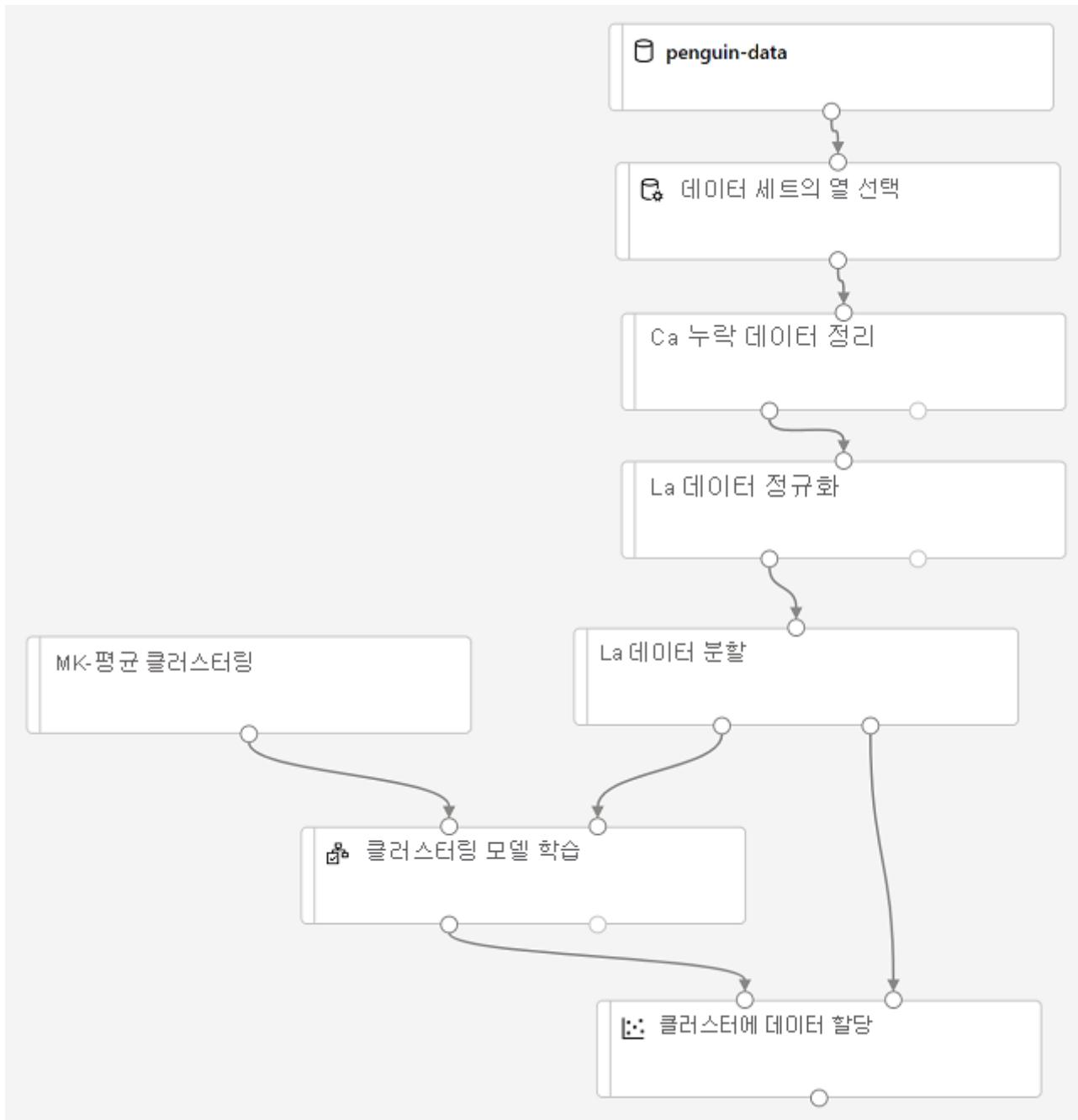
- K 좌표를 n 차원 공간의 중심 이라는 무작위로 선택된 지점으로 초기화합니다. 여기서 n 은 특징 벡터의 차원 수입니다.
- 특징 벡터를 동일한 공간의 지점으로, 각 지점을 가장 가까운 중심에 할당합니다.
- 중심을 그에 할당된 지점의 중앙으로 이동합니다(평균 거리를 기준으로 함).
- 이동 후에 가장 가까운 중심에 지점을 다시 할당합니다.

- e. 클러스터 할당이 안정화되거나 지정된 반복 횟수가 완료될 때까지 3단계와 4단계를 반복합니다.
-
8. 데이터의 70%를 사용하여 클러스터링 모델을 학습한 후에는 나머지 30%를 사용하여 데이터를 클러스터에 할당하는 모델을 사용함으로써 이를 테스트할 수 있습니다. **모델 채점 및 평가** 섹션을 확장하고 **클러스터에 데이터 할당** 모듈을 **클러스터링 모델 학습** 모듈 아래에 있는 캔버스로 끌어다 놓습니다. 그런 다음 **클러스터링 모델 학습**의 **학습된 모델**(왼쪽) 출력을 **클러스터에 데이터 할당** 모듈의 **학습된 모델**(왼쪽) 입력에 연결합니다. 그리고 **데이터 분할** 모듈의 결과 **데이터 세트2**(오른쪽) 출력을 **클러스터에 데이터 할당** 모듈의 **데이터 세트**(오른쪽) 입력에 연결합니다.

학습 파이프라인 실행

이제 학습 파이프라인을 실행하고 모델을 학습할 준비가 되었습니다.

1. 파이프라인이 다음과 같아야 합니다.



2. 제출 을 선택하고, 컴퓨팅 클러스터에서 **mslearn-penguin-training** 이라는 기존 실험을 사용하여 파이프라인을 실행합니다.
3. 실험이 완료될 때까지 기다립니다. 5분 이상 걸릴 수 있습니다.
4. 실험이 완료되면 **클러스터에 데이터 할당** 모듈을 선택하고 설정 창에서 **출력 + 로그** 탭의 **결과 데이터 세트** 섹션에 있는 **데이터 출력**에서 **시각화** 아이콘을 사용하여 결과를 봅니다.
5. 오른쪽으로 스크롤하여 각 행이 할당된 클러스터(0, 1 또는 2)를 포함하는 **할당** 열을 확인합니다. 또한 이 행을 나타내는 지점에서 각 클러스터의 중심까지의 거리를 나타내는 새 열이 있습니다. 지점과 가장 가까운 클러스터가 할당된 클러스터입니다.
6. **클러스터에 데이터 할당** 시각화를 닫습니다.

모델에서 펭귄 관찰에 대한 클러스터를 예측할 때 그 예측은 얼마나 안정적인가요? 이를 평가 하려면 모델을 평가해야 합니다.

< 이전

단위 6/10 ▼

다음 >

✓ 100 XP



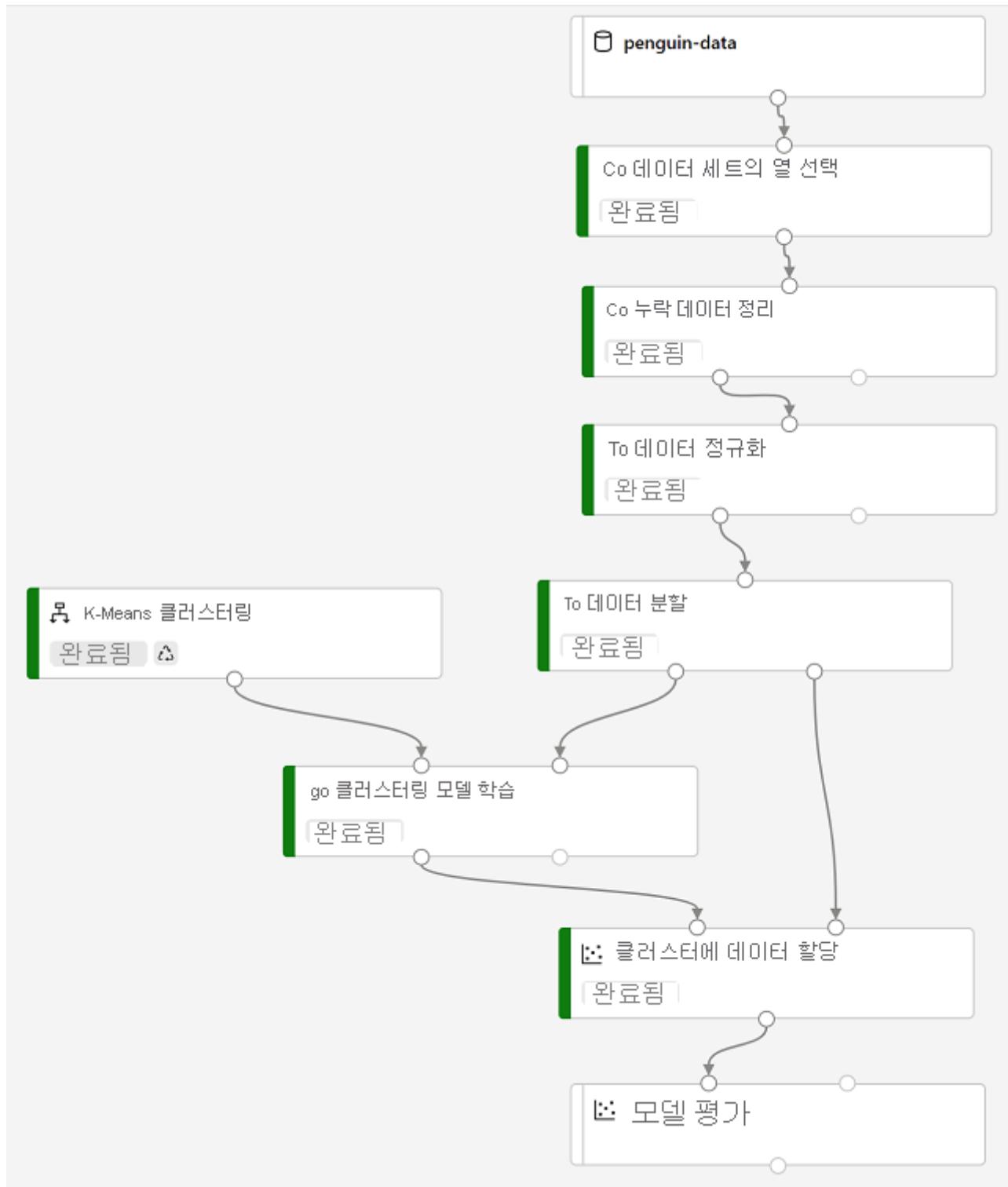
클러스터링 모델 평가

5분

클러스터링 모델을 평가하는 것은 클러스터 할당에 대해 이전에 알려진 *true* 값이 없으므로 어렵습니다. 성공적인 클러스터링 모델은 각 클러스터의 항목 간에 적절한 수준의 분리를 제공하는 모델이므로, 해당 분리를 측정하는 데 도움이 되는 메트릭이 필요합니다.

모델 평가 모듈 추가

1. 아직 열려 있지 않은 경우 이전 단원에서 만든 **Train Penguin Clustering** 파이프라인을 엽니다.
2. 왼쪽 창의 **모델 채점 및 평가** 섹션에서 **모델 평가** 모듈을 클러스터에 데이터 할당 모듈에 있는 캔버스로 끌어다 놓고, **클러스터에 데이터 할당** 모듈의 출력을 **모델 평가** 모듈의 **점수가 매겨진 데이터 세트(왼쪽)** 입력에 연결합니다.
3. 파이프라인이 다음과 같아야 합니다.



4. 제출을 선택하고, 기존 mslearn-penguin-training 실험을 사용하여 파이프라인을 실행합니다.
5. 실험이 완료될 때까지 기다립니다.
6. 실험이 완료되면 모델 평가 모듈을 선택하고 설정 창에서 출력 + 로그 탭의 평가 결과 섹션에 있는 데이터 출력에서 시각화 아이콘을 사용하여 성능 메트릭을 봅니다. 이러한 메트릭은 데이터 과학자가 모델이 클러스터를 얼마나 잘 분리하는지 평가하는 데 도움이 될 수 있습니다. 각 클러스터에 대한 메트릭 행과 결합된 평가의 요약 행을 포함합니다. 각 행의 메트릭은 다음과 같습니다.

- **다른 중심까지의 평균 거리:** 클러스터의 각 지점과 다른 모든 클러스터의 중심이 평균적으로 얼마나 가까운지 나타냅니다.
- **클러스터 중심까지의 평균 거리:** 클러스터의 각 지점과 클러스터의 중심이 평균적으로 얼마나 가까운지 나타냅니다.
- **지점 수:** 클러스터에 할당된 지점의 수입니다.
- **클러스터 중심까지의 최대 거리:** 각 지점과 해당 지점의 클러스터 중심 간 최대 거리입니다. 이 수치가 높으면 클러스터가 광범위하게 분산되었을 수 있습니다. **클러스터 중심까지의 평균 거리** 와 함께 이 통계를 활용하면 클러스터의 분산을 파악할 수 있습니다.

7. 모델 평가 결과 시각화 창을 닫습니다.

이제 작동하는 클러스터링 모델이 있으니 이를 사용하여 새 데이터를 추론 파이프라인에 있는 클러스터에 할당할 수 있습니다.

다음 단원: 유추 파이프라인 만들기

계속 >

< 이전

단위 7/10 ▼

다음 >

✓ 100 XP



유추 파이프라인 만들기

8분

파이프라인을 만들고 실행해 클러스터링 모델을 학습한 후에는 모델을 사용하여 클러스터에 새 데이터 관찰을 할당하는 추론 파이프라인을 만들 수 있습니다. 이를 통해 애플리케이션에서 사용하도록 게시할 수 있는 예측 서비스의 기반이 형성됩니다.

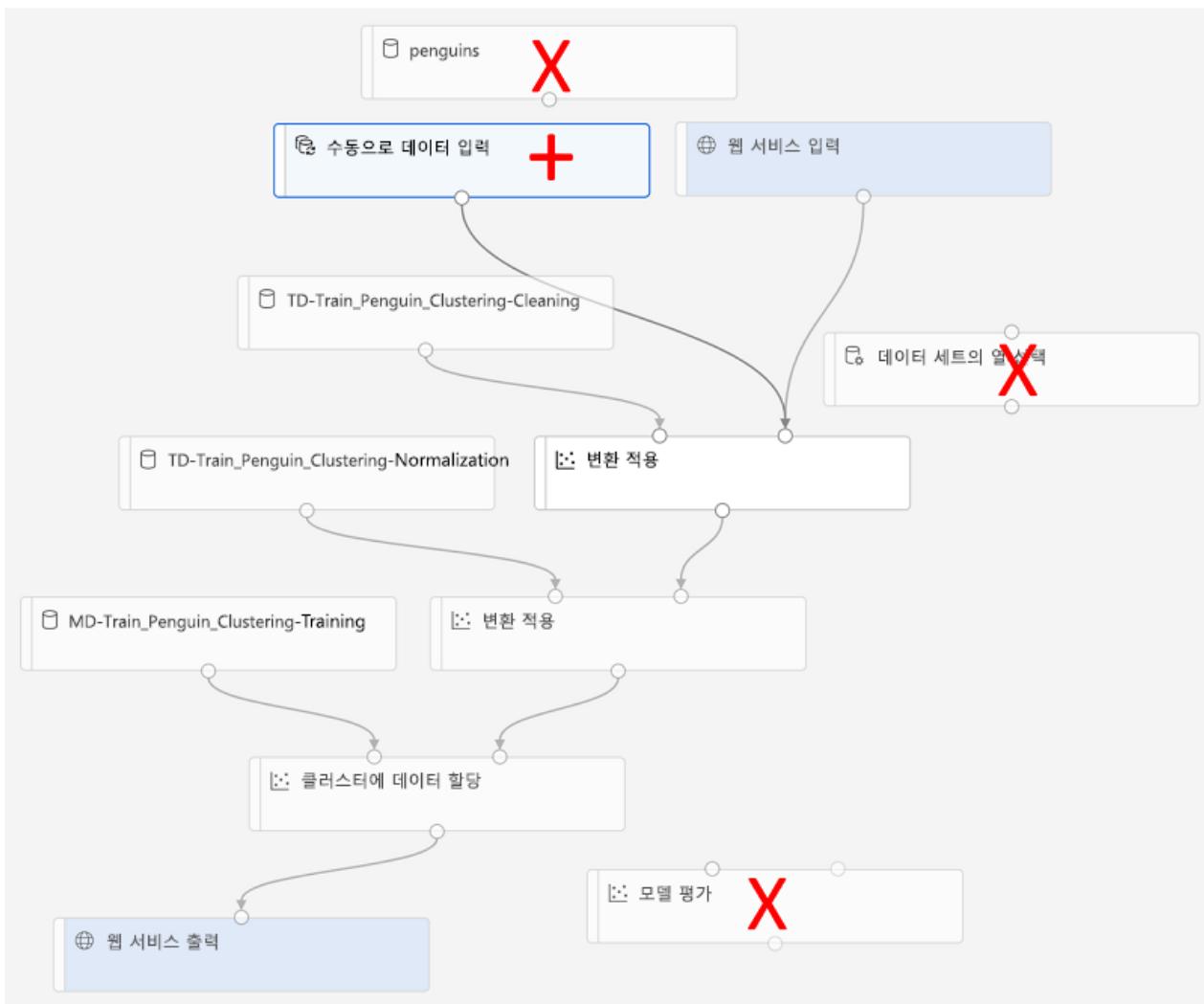
유추 파이프라인 만들기

1. Azure Machine Learning Studio에서 이전에 만든 **Train Penguin Clustering** 파이프라인을 엽니다.
2. **유추 파이프라인 만들기** 드롭다운 목록에서 **실시간 유추 파이프라인** 을 클릭합니다. 몇 초 후에 **Train Penguin Clustering-real time inference** 라는 새 버전의 파이프라인이 열립니다.

파이프라인에 웹 서비스 입력과 웹 서비스 출력 모듈이 포함되지 않은 경우 **디자이너 페** 이지로 돌아가 **Train Penguin Clustering-real time inference** 파이프라인을 다시 엽니다.

3. 새 파이프라인의 이름을 **Predict Penguin Clusters** 로 바꾸고 새 파이프라인을 검토합니다. 여기에는 제출할 새 데이터에 대한 웹 서비스 입력과 결과를 반환하기 위한 웹 서비스 출력이 포함되어 있습니다. 학습 파이프라인의 변환 및 클러스터링 모델은 학습 데이터의 통계를 기반으로 이 파이프라인에 캡슐화되며, 새 데이터의 변환 및 채점에 사용됩니다.

유추 파이프라인을 다음과 같이 변경합니다.



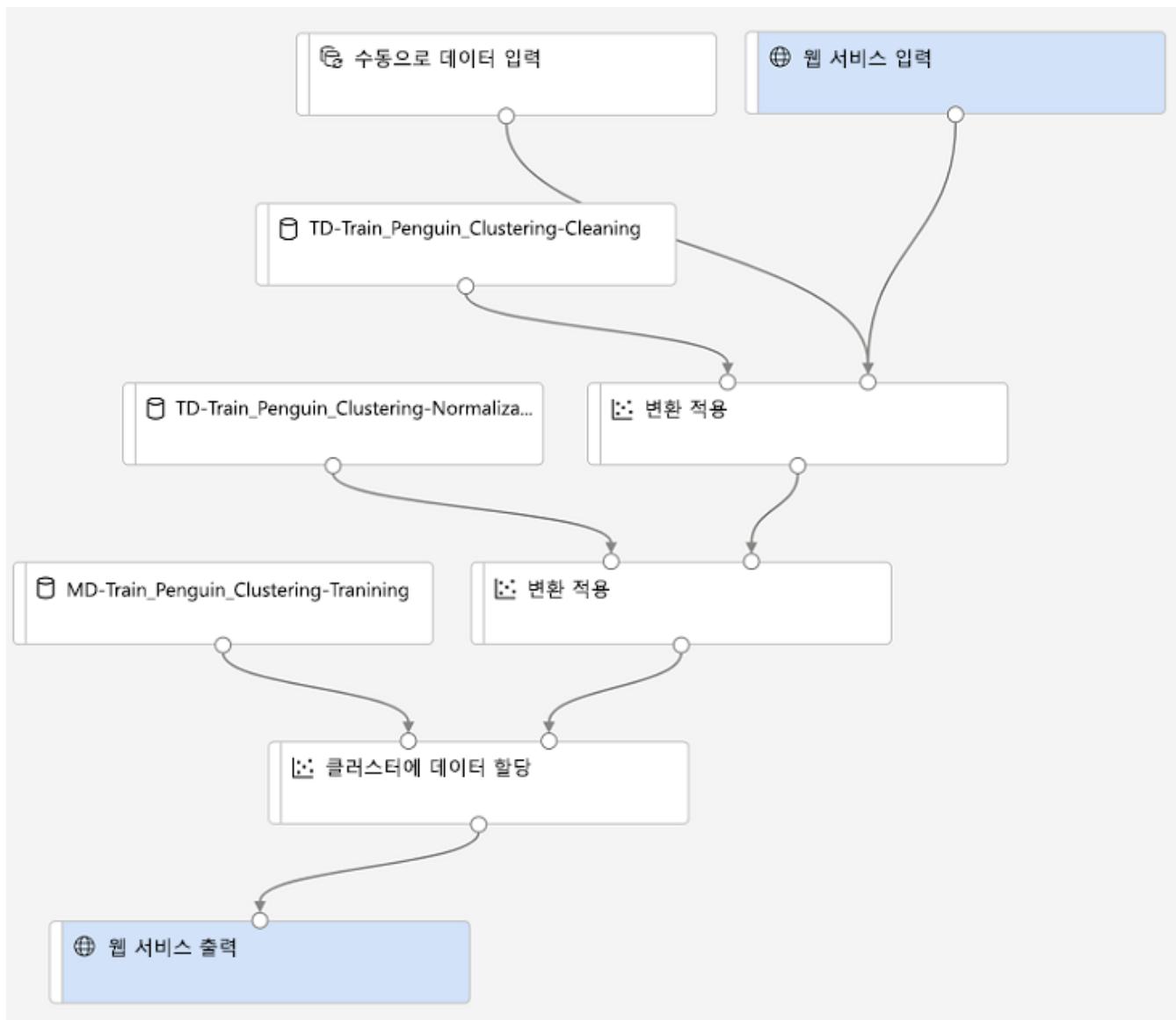
- penguin-data 데이터 세트를 Species 열을 포함하지 않은 수동으로 데이터 입력 모듈로 바꿉니다.
- 이제는 중복되는 데이터 세트에서 열 선택 모듈을 제거합니다.
- 웹 서비스 입력 및 수동으로 데이터 입력 모듈(클러스터링될 데이터의 입력을 나타냄)을 변환 적용 모듈에 연결합니다.
- 모델 평가 모듈을 제거합니다.

파이프라인을 수정할 때 위 이미지와 정보를 참조하여 아래의 나머지 단계를 수행 합니다.

4. 추론 파이프라인은 새 데이터가 원래 학습 데이터의 스키마와 일치하는 것으로 가정하므로 학습 파이프라인의 penguin-data 데이터 세트가 포함됩니다. 하지만 이 입력 데이터에는 모델에서 사용하지 않는 펭귄 종에 대한 열이 포함되어 있습니다. penguin-data 데이터 세트와 데이터 세트에서 열 선택 모듈을 모두 삭제하고 데이터 입력 및 출력 섹션에 있는 수동으로 데이터 입력 모듈로 대체합니다. 그런 다음 수동으로 데이터 입력 모듈의 설정을 수정하여 다음 CSV 입력을 사용합니다. 여기에는 3개의 새로운 펭귄 관찰에 대한 특징 값이 포함되어 있습니다(헤더 포함).

```
CulmenLength,CulmenDepth,FlipperLength,BodyMass
39.1,18.7,181,3750
49.1,14.8,220,5150
46.6,17.8,193,3800
```

5. 웹 서비스 입력 및 수동으로 데이터 입력 모듈의 출력을 첫 번째 변환 적용 모듈의 데이터 세트(오른쪽) 입력에 연결합니다.
6. 모델 평가 모듈을 삭제합니다.
7. 파이프라인이 다음과 유사한지 확인합니다.



8. 컴퓨팅 클러스터에서 `mslearn-penguin-inference`라는 새 실험으로 파이프라인을 제출합니다. 이 작업은 다소 시간이 걸릴 수 있습니다.
9. 파이프라인이 완료되면 **클러스터에 데이터 할당** 모듈의 결과 데이터 세트 출력을 시각화하여 입력 데이터의 3가지 펭귄 관찰에 대한 예측된 클러스터 할당 및 메트릭을 확인합니다.

추론 파이프라인은 특징을 기반으로 펭귄 관찰을 클러스터에 할당합니다. 이제 클라이언트 애플리케이션에서 사용할 수 있도록 파이프라인을 게시할 준비가 되었습니다.

다음 단원: 예측 서비스 배포

계속 >

[〈 이전](#)

단위 8/10 ▼

[다음 〉](#)

✓ 100 XP



예측 서비스 배포

5분

실시간 추론용 추론 파이프라인을 만들고 테스트한 후에는 이를 클라이언트 응용 프로그램에서 사용할 서비스로 게시할 수 있습니다.

참고: 이 연습에서는 ACI(Azure Container Instance)에 웹 서비스를 배포합니다. 이러한 유형의 컴퓨팅은 동적으로 만들어지며 개발 및 테스트에 유용합니다. 프로덕션 환경에서는 '유추 클러스터'를 만들어 향상된 스케일링 성능 및 보안을 제공하는 AKS(Azure Kubernetes Service) 클러스터를 제공해야 합니다.

서비스 배포

1. 이전 단원에서 만든 **Predict Penguin Clusters** 추론 파이프라인을 봅니다.
2. 오른쪽 위에서 **배포**를 선택하고 다음 설정을 사용하여 새로운 실시간 엔드포인트를 배포합니다.
 - **이름:** predict-penguin-clusters
 - **설명:** 펭귄 클러스터
 - **컴퓨팅 형식:** Azure Container Instances
3. 웹 서비스가 배포될 때까지 기다립니다. 몇 분 정도 걸릴 수 있습니다. 배포 상태는 디자이너 인터페이스의 왼쪽 상단에 표시됩니다.

서비스 테스트

이제 클라이언트 애플리케이션에서 배포된 서비스를 테스트할 수 있습니다. 여기에서는 아래 셀의 코드를 사용하여 클라이언트 애플리케이션을 시뮬레이션합니다.

1. **엔드포인트** 페이지에서 **predict-penguin-clusters** 실시간 엔드포인트를 엽니다.
2. **predict-penguin-clusters** 엔드포인트가 열리면 **사용** 탭을 보고 다음 정보를 확인합니다. 클라이언트 애플리케이션에서 배포된 서비스로 연결하려면 해당 정보가 필요합니다.
 - 서비스에 대한 REST 엔드포인트
 - 서비스에 대한 기본 키
3. 이 값 옆에 있는 링크를 사용하여 이를 클립보드에 복사할 수 있습니다.

4. 브라우저에서 **predict-penguin-clusters** 서비스 페이지의 사용 페이지가 열린 상태에서 새 브라우저 탭을 열고 **Azure Machine Learning Studio** 의 두 번째 인스턴스를 엽니다. 그런 다음, 새 탭에서 **Notebooks** 페이지(작성자 아래)를 봅니다.
5. **Notebooks** 페이지의 **내 파일**에서 □ 단추를 사용하여 다음 설정으로 새 파일을 만듭니다.
 - **파일 위치:** Users/사용자 이름
 - **파일 이름:** Test-Penguins
 - **파일 형식:** Notebook
 - **이미 있는 경우 덮어쓰기:** 선택됨
6. 새 Notebook이 만들어지면 이전에 만든 컴퓨팅 인스턴스가 **컴퓨팅** 상자에서 선택되었는지, 그리고 상태가 **실행 중**인지 확인합니다.
7. << 단추를 사용하여 파일 탐색기 창을 축소하고 **Test-Penguins.ipynb** Notebook 탭에 사용할 수 있는 공간을 확보합니다.
8. Notebook에서 만들어진 사각형 셀에 다음 코드를 붙여넣습니다.

Python

복사

```

endpoint = 'YOUR_ENDPOINT' #Replace with your endpoint
key = 'YOUR_KEY' #Replace with your key

import urllib.request
import json
import os

data = {
    "Inputs": {
        "WebServiceInput0": [
            {
                "CulmenLength": 49.1,
                "CulmenDepth": 4.8,
                "FlipperLength": 1220,
                "BodyMass": 5150,
            },
        ],
    },
    "GlobalParameters": {}
}

body = str.encode(json.dumps(data))

headers = {'Content-Type': 'application/json', 'Authorization':('Bearer ' +
key)}

req = urllib.request.Request(endpoint, body, headers)

```

```
try:  
    response = urllib.request.urlopen(req)  
    result = response.read()  
    json_result = json.loads(result)  
    output = json_result["Results"]["WebServiceOutput0"][0]  
    print('Cluster: {}'.format(output["Assignments"]))  
  
except urllib.error.HTTPError as error:  
    print("The request failed with status code: " + str(error.code))  
  
    # Print the headers to help debug  
    print(error.info())  
    print(json.loads(error.read().decode("utf8", 'ignore')))
```

① 참고

코드의 세부 정보에 대해 너무 걱정하지 마세요. 코드는 Penguin에 대한 기능을 정의하고, 만든 **predict-penguin-clusters** 서비스를 사용하여 클러스터 할당을 예측할 뿐입니다.

9. **predict-penguin-clusters** 서비스의 사용 페이지가 포함된 브라우저 탭으로 전환한 다음 서비스에 대한 REST 엔드포인트를 복사합니다. Notebook을 포함하는 탭으로 돌아가 키를 코드에 복사하고 YOUR_ENDPOINT를 대체합니다.
10. **predict-penguin-clusters** 서비스의 사용 페이지가 포함된 브라우저 탭으로 전환한 다음 서비스에 대한 기본 키를 복사합니다. Notebook을 포함하는 탭으로 돌아가 키를 코드에 복사하고 YOUR_KEY를 대체합니다.
11. Notebook을 저장한 다음 셀 옆에 있는 ▶ 단추를 사용하여 코드를 실행합니다.
12. 예측된 클러스터가 반환되는지 확인합니다.

다음 단원: 지식 점검

계속 >



Microsoft Azure에서 컴퓨터 비전 살펴보기

3600 XP

2시간 53분 • 학습 경로 • 6 모듈

초급 AI 엔지니어 데이터 과학자 개발자 솔루션 아키텍처 학생

Cognitive Services

컴퓨터 비전은 소프트웨어 시스템이 카메라, 이미지 및 비디오를 통해 실제 대상을 시각적으로 인지하도록 설계된 AI(인공 지능)의 한 분야입니다. AI 엔지니어와 데이터 과학자는 Microsoft Azure의 여러 인지 서비스를 비롯한 PaaS(Platform as a Service) 솔루션과 사용자 지정 기계 학습 모델을 조합하여 여러 구체적인 유형의 컴퓨터 비전 문제를 해결할 수 있습니다.

필수 조건

Azure Portal을 탐색할 수 있는 능력

책갈피 컬렉션에 추가

이 학습 경로의 모듈



Computer Vision 서비스를 사용하여 이미지 분석

600 XP

28분 • 모듈 • 5 단위

4.7 (897)

Computer Vision 서비스를 사용하면 소프트웨어 엔지니어가 여러 인공 지능(AI) 시나리오에 공통된 작업인 이미지로부터 정보 추출을 수행하는 지능형 솔루션을 만들 수 있습니다.

개요



Custom Vision 서비스를 사용한 이미지 분류

34분 • 모듈 • 5 단위

★★★★★ ★★ 4.8 (760)

이미지 분류는 AI(인공 지능) 애플리케이션에서 일반적인 작업입니다. 기계 학습의 예측 능력을 사용하여 AI 시스템이 이미지에 기반한 실제 항목을 식별할 수 있도록 합니다.

개요 ▾



Custom Vision 서비스를 사용하여 이미지의 개체 감지

✓ 600 XP



39분 • 모듈 • 5 단위

★★★★★ ★★ 4.8 (699)

개체 감지는 AI(인공 지능) 에이전트가 이미지 또는 카메라 피드에서 특정 유형의 개체를 식별하고 찾을 수 있는 Computer Vision의 한 형태입니다.

개요 ▾



Face 서비스를 통해 얼굴 감지 및 분석

✓ 600 XP



24분 • 모듈 • 5 단위

★★★★★ ★★ 4.8 (667)

얼굴 감지, 분석 및 인식은 AI(인공 지능) 솔루션의 중요한 기능입니다. Azure의 Face 인식 서비스를 사용하면 이러한 기능을 애플리케이션에 쉽게 통합할 수 있습니다.

개요 ▾



Computer Vision 서비스로 텍스트 읽기

✓ 600 XP



24분 • 모듈 • 5 단위



★★★★★ 4.8 (608)

OCR(광학 인식)을 통해 AI(인공 지능) 시스템에서는 이미지의 텍스트를 읽을 수 있으므로 애플리케이션에서 사진, 스캔한 문서 및 기타 디지털 텍스트 원본에서 정보를 추출할 수 있습니다.

개요 ▾



Form Recognizer 서비스를 사용하여 영수증 분석

✓ 600 XP



24분 • 모듈 • 5 단위

★★★★★ 4.8 (792)

청구서 및 영수증 처리는 많은 비즈니스 시나리오에서 일반적인 작업입니다. 점점 더 많은 조직이 AI(인공 지능)를 활용하여 스캔한 영수증으로부터의 데이터 추출을 자동화하고 있습니다.

개요 ▾





✓ 600 XP ➔

Computer Vision 서비스를 사용하여 이미지 분석

28분 • 모듈 • 5 단위

★★★★★ 4.7 (897)

평가하기

초급 AI 엔지니어 데이터 과학자 개발자 솔루션 아키텍처 학생

Cognitive Services

Computer Vision 서비스를 사용하면 소프트웨어 엔지니어가 여러 인공 지능(AI) 시나리오에 공통된 작업인 이미지로부터 정보 추출을 수행하는 지능형 솔루션을 만들 수 있습니다.

학습 목표

Computer Vision 인지 서비스를 사용하여 이미지를 분석하는 방법을 알아봅니다.

📘 책갈피 Ⓛ 컬렉션에 추가

필수 조건

Azure Portal 탐색 능력

이 모듈은 이러한 학습 경로의 일부입니다.

[Microsoft Azure에서 컴퓨터 비전 살펴보기](#)

소개

2분



Azure에서 이미지 분석 시작

3분



연습 - Computer Vision 서비스를 사용하여 이미지 분석

20분



지식 점검

2분



요약

1분



✓ 100 XP



소개

2분

*Computer Vision*은 AI(인공 지능)의 핵심 영역 중 하나이며 AI 지원 애플리케이션이 세상을 "보고" 이해할 수 있는 솔루션을 만드는 데 중점을 둡니다.

컴퓨터에는 당연히 인간과 같은 생물학적 눈이 없지만 라이브 카메라 피드나 디지털 사진 또는 비디오로 이미지를 처리할 수 있습니다. 이미지를 처리하는 이 기능은 인간의 시각적 인식을 모방할 수 있는 소프트웨어를 만드는 핵심입니다.

AI 애플리케이션에서 이미지는 픽셀 값의 배열일 뿐입니다. 이러한 숫자 값은 이미지와 그 콘텐츠를 예측하는 기계 학습 모델을 학습시키는 특징으로 사용할 수 있습니다.

Microsoft Azure에서 **Computer Vision** 인지 서비스는 미리 학습된 모델을 사용하여 이미지를 분석하므로 소프트웨어 개발자는 다음을 수행하는 애플리케이션을 빌드할 수 있습니다.

- 이미지를 해석하고 적절한 캡션을 제안합니다.
- 이미지를 인덱싱하는 데 사용할 수 있는 관련 태그를 제안합니다.
- 이미지를 분류합니다.
- 이미지에서 개체를 식별합니다.
- 이미지에서 얼굴 및 사람을 감지합니다.
- 이미지에서 유명인 및 랜드마크를 인식합니다.
- 이미지의 텍스트를 읽습니다.

이 모듈에서는 Computer Vision 서비스를 사용하여 이미지를 분석하는 방법과 이러한 기능 중 일부를 살펴보겠습니다.

다음 단원: Azure에서 이미지 분석 시작

[계속 >](#)

[〈 이전](#)

단위 2/5 ▼

[다음 〉](#)

✓ 100 XP



Azure에서 이미지 분석 시작

3분

Computer Vision 서비스는 미리 빌드된 Computer Vision 기능을 제공하는 Microsoft Azure의 인지 서비스입니다. 서비스는 이미지를 분석하고 이미지 및 이미지에서 묘사되는 개체에 대한 세부 정보를 반환할 수 있습니다.

Computer Vision용 Azure 리소스

Computer Vision 서비스를 사용하려면 Azure 구독에서 리소스를 만들어야 합니다. 다음 리소스 유형 중 하나를 사용할 수 있습니다.

- **Computer Vision** : Computer Vision 서비스에 대한 특정 리소스입니다. 다른 인지 서비스를 사용하지 않으려는 경우 또는 Computer Vision 리소스의 사용률과 비용을 별도로 추적 하려는 경우 이 리소스 유형을 사용하세요.
- **Cognitive Services** : Text Analytics 및 Translator Text 등의 다른 많은 인지 서비스와 함께 Computer Vision을 포함하는 일반적인 인지 서비스 리소스입니다. 여러 인지 서비스를 사용할 계획이며 관리 및 개발을 단순화하려는 경우 이 리소스 유형을 사용하세요.

만들려는 리소스 유형과 관계없이 사용을 위해 필요한 두 가지 정보가 제공됩니다.

- 클라이언트 애플리케이션을 인증하는 데 사용되는 **키**입니다.
- 리소스에 액세스할 수 있는 HTTP 주소를 제공하는 **엔드포인트**입니다.

ⓘ 참고

Cognitive Services 리소스를 만드는 경우 클라이언트 응용 프로그램에서는 사용 중인 특정 서비스와 상관없이 동일한 키와 엔드포인트를 사용합니다.

Computer Vision Service를 사용하여 이미지 분석

구독에서 적합한 리소스를 만든 후에는 다양한 분석 작업을 수행할 수 있도록 이미지를 Computer Vision 서비스에 제출할 수 있습니다.

이미지 설명

Computer Vision은 이미지를 분석하고, 감지된 개체를 평가하고, 사용자가 읽을 수 있는 구문이나 문장을 생성하여 이미지에서 감지된 내용을 설명할 수 있습니다. 이미지 콘텐츠에 따라 서비스는 여러 개의 결과 또는 구문을 반환할 수 있습니다. 반환된 각 구문에는 해당하는 신뢰도 점수가 있어 제공된 설명에서 알고리즘의 신뢰도를 확인할 수 있습니다. 신뢰도가 가장 높은 구문부터 먼저 나열됩니다.

이 개념을 이해하는 데 도움이 되도록 예를 들어 보겠습니다. 다음 뉴욕 엠파이어 스테이트 빌딩의 이미지를 살펴보세요. 반환된 구문은 신뢰도 순으로 이미지 아래에 나열됩니다.



- 흑백 도시 사진
- 흑백 대도시 사진
- 도시의 커다란 하얀 건물

시각적 특징 태그 지정

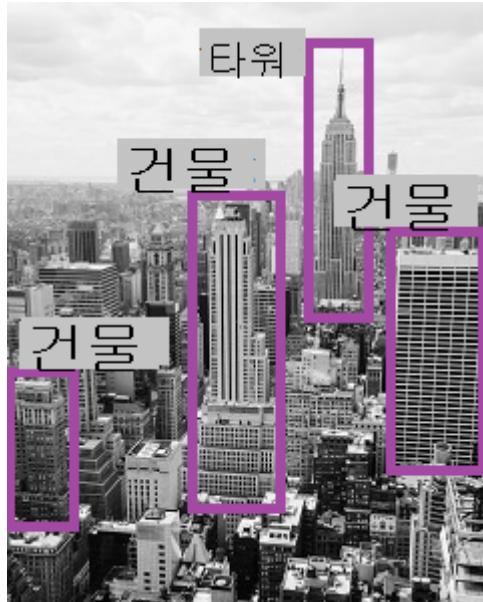
Computer Vision에서 생성되는 이미지 설명은 수천 개의 인식 가능한 개체 집합을 기반으로 하며, 이를 사용하여 이미지에 대한 태그를 제안하는 데 사용할 수 있습니다. 이 태그는 이미지의 특성을 요약하는 메타데이터로서 이미지와 연결될 수 있습니다. 특히 특정 특성 또는 콘텐츠를 포함한 이미지를 검색하는 데 사용할 수 있는 주요 용어 집합과 함께 이미지를 인덱싱하고자 하는 경우 유용합니다.

예를 들어 엠파이어 스테이트 빌딩 이미지에 대해 반환되는 태그는 다음을 포함합니다.

- 고층 건물
- 타워
- 건물

개체 감지

개체 감지 기능은 서비스에서 공통 개체를 식별할 수 있다는 점에서 태그 지정과 비슷합니다. 하지만 태그를 지정하거나 인식된 개체에 대해서만 태그를 제공하는 대신, 이 서비스는 경계 상자 좌표를 반환할 수도 있습니다. 개체 감지는 개체의 형식을 가져올 뿐만 아니라 감지된 개체의 상단, 왼쪽, 너비 및 높이를 나타내는 좌표 집합도 제공하며, 이는 다음과 같이 이미지에서 개체의 위치를 식별하는 데 사용할 수 있습니다.



브랜드 감지

이 기능은 상용 브랜드를 식별하는 기능을 제공합니다. 서비스에는 전 세계적으로 인식된 수천 개의 상용 제품 브랜드 로고가 포함된 데이터베이스가 있습니다.

서비스를 호출하고 이미지를 전달하면 이 기능은 감지 작업을 수행하고 이미지의 식별된 개체가 인식된 브랜드인지 확인합니다. 서비스는 의류, 전자 제품 등 다양한 범주에 속하는 인기 브랜드 데이터베이스와 브랜드를 비교합니다. 알려진 브랜드가 감지되면 서비스는 브랜드 이름, 신뢰도 점수(0~1, 식별 결과의 긍정도를 나타냄), 이미지에서 감지된 브랜드가 있는 위치의 경계 상자(좌표)를 반환합니다.

예를 들어 다음 이미지의 노트북 덮개에는 Microsoft 로고가 있으며, Computer Vision 서비스에서 이를 식별했습니다.

'Microsoft' (Confidence: 68.00%)



얼굴 감지

Computer Vision 서비스는 이미지에서 사람의 얼굴을 감지 및 분석할 수 있습니다. 여기에는 나아가 얼굴의 위치를 나타내는 사각형 경계 상자의 위치를 결정하는 기능 또한 포함되어 있습니다. Computer Vision 서비스의 얼굴 분석 기능은 전용 [Face 서비스](#)에서 제공하는 하위 기능입니다. 일반 이미지 분석 기능과 결합된 기본적인 얼굴 감지 및 분석이 필요한 경우 Computer Vision 서비스를 사용할 수 있습니다. 하지만 더욱 포괄적인 얼굴 분석 및 얼굴 인식 기능을 사용하려면 Face 서비스를 사용해야 합니다.

다음 예제에서는 사람의 이미지에서 얼굴을 감지하고 대략적인 나이를 추정하는 것을 보여줍니다.



이미지 분류

Computer Vision은 콘텐츠를 기준으로 이미지를 분류할 수 있습니다. 서비스는 "현재" 제한되어 있는 범주 집합을 포함한 부모/자식 계층 구조를 사용합니다. 이미지를 분석할 때 감지된 개체가 기준 범주와 비교되어 분류를 제공하는 최적의 방법이 결정됩니다. 부모 범주 중 하나로는 **people_** 이 있습니다. 지붕 위에 있는 사람의 이미지에는 **people_** 범주가 할당됩니다.



다음 이미지에 대해서는 약간 다른 분류가 반환되는데, 이미지 안에 여러 명의 사람이 있으므로 **people_group** 범주가 할당됩니다.



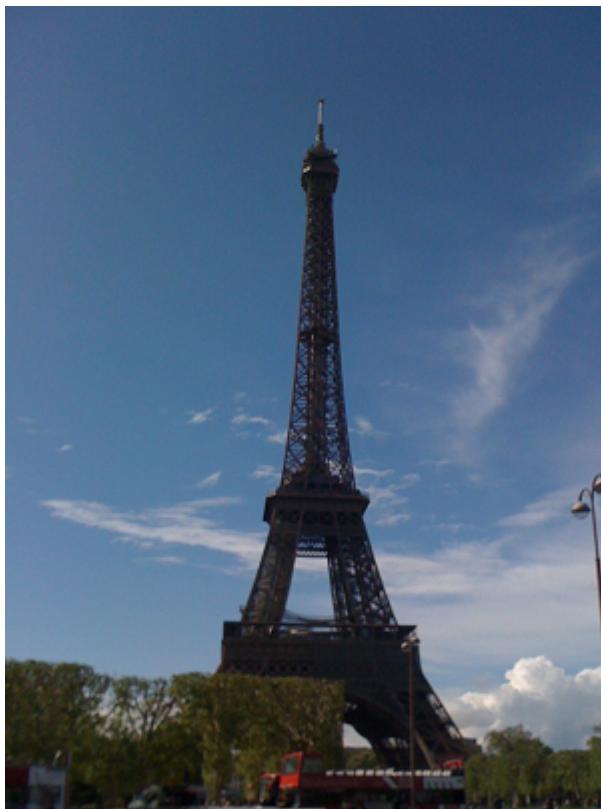
[여기](#)에서 86개의 범주 목록을 검토하세요.

도메인별 콘텐츠 감지

이미지를 분류할 때 Computer Vision 서비스는 다음과 같은 두 가지 특수 도메인 모델을 지원합니다.

- **유명인** - 이 서비스에는 스포츠, 엔터테인먼트 및 비즈니스 분야에서 잘 알려진 수천 명의 유명인을 식별하도록 학습된 모델이 포함되어 있습니다.
- **랜드마크** - 이 서비스는 타지마할 및 자유의 여신상 같은 유명 랜드마크를 식별할 수 있습니다.

예를 들어 다음 이미지에서 랜드마크를 분석할 때 Computer Vision 서비스는 에펠탑을 99.41%의 신뢰도로 식별합니다.



광학 인식

Computer Vision 서비스는 OCR(광학 인식) 기능을 사용하여 이미지에서 인쇄 및 필기 텍스트를 감지할 수 있습니다. 이 기능에 대해 살펴보려면 Microsoft Learn의 [Computer Vision 서비스로 텍스트 읽기 모듈](#)을 참조하세요.

추가 기능

이러한 기능 외에도 Computer Vision 서비스는 다음을 수행할 수 있습니다.

- 이미지 유형 감지 - 클립 아트 이미지 또는 선 그리기를 식별합니다.
- 이미지 색 구성표 감지 - 특히 이미지의 주요 전경, 배경 및 전체 색을 식별합니다.
- 썸네일 생성 - 이미지의 작은 버전을 생성합니다.
- 일반 콘텐츠 - 성인용 콘텐츠를 포함하거나 잔인한 장면을 묘사하는 이미지를 감지합니다.

다음 단원: 연습 - Computer Vision 서비스를 사용하여 이미지 분석

계속 >

< 이전

단위 3/5 ▼

다음 >

100 XP

연습 - Computer Vision 서비스를 사용하여 이미지 분석

20분

Computer Vision 서비스의 기능을 학습하는 가장 좋은 방법은 직접 살펴보는 것입니다.

시작하기 전에

이 연습을 완료하려면 다음이 필요합니다.

- Microsoft Azure 구독. 구독이 아직 없다면 <https://azure.microsoft.com/free>에서 평가판에 가입할 수 있습니다.
- MicrosoftDocs/ai-fundamentals GitHub 리포지토리에서 Jupyter Notebook을 실행할 수 있는 Python 환경. GitHub의 랩 파일을 자신의 Python 환경(있는 경우)에 복제하거나, 아래 지침에 따라 Azure 구독에서 Azure Machine Learning 작업 영역을 만들 수 있습니다.

ⓘ 참고

이 모듈은 Azure Machine Learning 작업 영역을 사용하는 여러 가지 모듈 중 하나입니다. **Azure AI 기본 사항** 인증의 준비 과정에서 이 모듈을 완료하는 경우, 작업 영역을 한 번 만들고 해당 인증 트랙의 모든 모듈에 다시 사용할 수 있습니다. 연습을 완료한 후에는 정리 지침에 따라 모듈 간 컴퓨팅 리소스를 중지해야 합니다.

Azure Machine Learning 작업 영역 만들기

Azure 구독에 Azure Machine Learning 작업 영역이 이미 있는 경우 새 브라우저 탭에서 [Azure Machine Learning Studio](#)로 이동하고 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다. 없는 경우 다음 단계에 따라 새 작업 영역을 만듭니다.

1. Azure 구독과 연결된 Microsoft 계정을 사용하여 [Azure Portal](#)에 로그인합니다.
2. + 리소스 만들기를 선택하고, *Machine Learning*을 검색한 뒤 다음 설정을 사용하여 새 *Machine Learning* 리소스를 만듭니다.
 - **작업 영역 이름:** 원하는 고유한 이름을 입력
 - **구독:** 자신의 Azure 구독
 - **리소스 그룹:** 고유한 이름의 새 리소스 그룹 만들기
 - **위치:** 사용 가능한 위치 선택

3. 작업 영역 리소스가 만들어질 때까지 기다립니다(몇 분이 걸릴 수 있음). 그런 다음 포털로 이동하고 작업 영역의 **개요** 페이지에서 Azure Machine Learning Studio를 실행(또는 새 브라우저 탭을 열고 <https://ml.azure.com> 으로 이동)한 다음 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다.
4. Azure Machine Learning Studio에서 왼쪽 상단에 있는 **☰** 아이콘을 설정/해제하면 인터페이스에 다양한 페이지가 표시됩니다. 해당 페이지를 사용하여 작업 영역에서 리소스를 관리할 수 있습니다.

컴퓨팅 인스턴스 만들기

이 연습에서 사용된 Notebook을 실행하려면 Azure Machine Learning 작업 영역에 컴퓨팅 인스턴스가 필요합니다. 컴퓨팅 인스턴스가 이미 있는 경우 시작하고, 없는 경우 다음 지침에 따라 새로 만듭니다.

1. [Azure Machine Learning Studio](#)에서 **관리** 아래에 있는 **컴퓨팅** 페이지를 확인합니다.
2. **컴퓨팅 인스턴스** 탭에서 다음 설정을 사용하여 새 컴퓨팅 인스턴스를 만듭니다.
 - **지역**: 사용 가능한 지역 선택
 - **가상 머신 유형**: CPU
 - **가상 머신 크기**: Standard_DS11_v2
 - **컴퓨팅 이름**: 고유한 이름 입력
3. 컴퓨팅 인스턴스가 시작될 때까지 기다립니다(1분 정도 걸릴 수 있음).

연습 파일 다운로드

이 모듈(및 기타 관련 모듈)에 사용되는 파일은 GitHub 리포지토리에 게시되는데, 이를 자신의 Python 환경에 복제해야 합니다. 이전 모듈에서 **ai-fundamentals** 리포지토리를 아직 복제하지 않은 경우 다음 단계를 사용하여 Azure Machine Learning 작업 영역에 복제합니다.

1. [Azure Machine Learning Studio](#)에서 **Notebooks** 페이지(**작성자** 아래)를 확인합니다. 이 페이지에는 Notebooks 실행에 사용할 수 있는 Notebook 편집기가 포함되어 있습니다.
2. **내 파일**에서 **□** 단추를 사용하여 다음 설정에 따라 새 파일을 만듭니다.
 - **파일 위치**: `Users/your user name`
 - **파일 이름**: Get-Files
 - **파일 형식**: Notebook
 - **이미 있는 경우 덮어쓰기**: 선택됨
3. 새 Notebook이 만들어지면 이전에 만든 컴퓨팅 인스턴스가 **컴퓨팅** 상자에서 선택되었는지, 그리고 상태가 **실행 중**인지 확인합니다. 그런 다음 Notebook에서 만든 사각형 셀에 다음 코드를 붙여넣습니다.

```
!git clone https://github.com/MicrosoftDocs/ai-fundamentals
```

4. 셀 옆에 있는 ▶ 단추를 사용하여 포함된 코드를 실행합니다. 이렇게 하면 GitHub에서 연습 파일이 복제됩니다.
5. 코드의 실행이 끝나고 파일의 체크아웃이 완료되면 내 파일 아래에 있는 ⌂ 단추를 사용하여 폴더 보기를 새로 고치고, **ai-fundamentals**라는 이름의 폴더가 만들어졌는지 확인합니다. 이 폴더에는 연습에 사용된 Notebook 및 기타 파일이 포함되어 있습니다.
6. Get-Files.ipynb Notebook 탭을 닫습니다.

연습 완료

Python 환경을 설정하고 **ai-fundamentals** 리포지토리를 복제했으면 Computer Vision에서 이미지 분석을 살펴볼 준비가 된 것입니다.

1. **ai-fundamentals** 폴더에서 **Image Analysis with Computer Vision.ipynb** Notebook을 엽니다. Azure Machine Learning Studio에서 Notebook 편집기를 사용하는 경우, <> 단추로 파일 탐색기 창을 축소하여 Notebook 탭에 집중할 수 있는 공간을 좀 더 마련합니다.
2. Notebook의 정보를 읽고 포함된 코드 셀을 순서대로 실행합니다.

정리

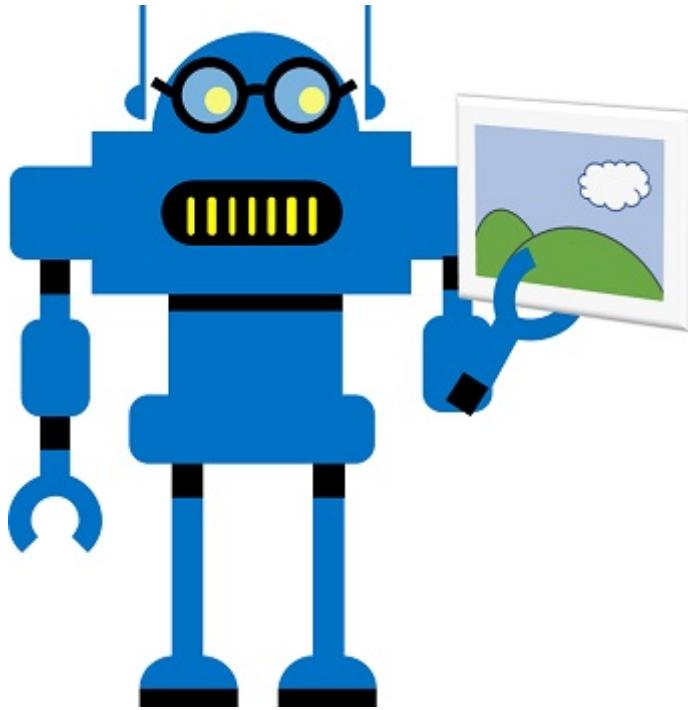
Azure Machine Learning Studio에서 컴퓨팅 인스턴스를 사용한 경우 Azure 크레딧을 불필요하게 사용하지 않도록 이를 중지해야 합니다.

1. Azure Machine Learning Studio에서 관리 아래에 있는 컴퓨팅 페이지를 확인합니다.
2. 컴퓨팅 인스턴스 탭에서 컴퓨팅 인스턴스를 선택하고 중지 단추를 사용하여 중지합니다.

다음 단원: 지식 점검

계속 >

Computer Vision 서비스로 이미지 분석하기



*Computer Vision*은 인공지능(AI)의 한부분으로써 실시간으로 카메라를 이용하거나 이미지나 비디오를 분석하는 등 "볼"수 있는 인공지능 시스템 개발을 하는 것이다. 이것은 디지털 이미지가 본질적으로 단지 숫자로된 픽셀의 값으로 이루어져 있음으로 해서 가능하며 우리는 이 픽셀값을 *features*로 사용하여 머신러닝 모델로 하여금 학습을 하여 이미지를 분류하고, 하나의 이미지에서 객체를 찾아내고 심지어는 사진에 대한 텍스트 기반의 요약정보를 만들어 낼 수 있다.

Computer Vision Cognitive 서비스 사용하기

Microsoft Azure에는 일반적인 AI 기능을 캡슐화하는 *Cognitive* 서비스가 포함되어 있으며, 여기에는 컴퓨터 비전 솔루션을 구축하는 데 도움이 될 수 있는 기능도 포함되어 있습니다.

Computer Vision Cognitive 서비스는 Azure에서 컴퓨터 비전 팀사를 위한 명확한 출발점을 제공합니다. 사전 훈련된 기계 학습 모델을 사용하여 이미지를 분석하고 이에 대한 정보를 추출한다.

예를 들어 Northwind Traders가 AI 서비스가 매장을 모니터링해 지원이 필요한 고객을 파악하고 직원들에게 도움을 지시하는 '스마트 스토어'를 구현하기로 했다고 가정해보자. 컴퓨터 비전 서비스를 이용하면 매장 곳곳에서 카메라가 촬영한 영상을 분석해 그들이 묘사한 내용에 대한 의미 있는 설명을 제공할 수 있다..

Cognitive Services 리소스 생성하기

Azure subscription에서 **Cognitive Services**를 추가하는 것으로 시작하자.

1. 브라우저의 새로운 탭을 열고, Azure portal(<https://portal.azure.com>)을 (<https://portal.azure.com>)을) 입력하고, Microsoft 계정으로 로그인한다..
2. + 리소스 만들기 버튼을 클릭하고, *Cognitive Services* 서비스를 찾은 다음, **Cognitive Services** 리소스를 다음과 같은 내용으로 생성한다.
 - **이름:** 유일한 이름을 입력한다(가능하면 영문과 숫자 사용).
 - **구독:** Azure 구독선택.
 - **위치:** 가능한 위치(한국 중부 추천):
 - **가격책정계층:** 표준 S0

- **리소스 그룹:** 원하는 유리한 이름(가능하면 영문과 숫자사용).
3. 배포가 완료될 때까지 기다린다. 그런 다음 Cognitive Services 리소스로 이동하여 *개요 페이지에서 링크를 클릭하여 서비스 키를 관리한다. 클라이언트 응용 프로그램에서 Cognitive Services 리소스에 연결하려면 엔드포인트와 키가 필요하다.

Cognitive Services 리소스에 있는 키와 엔드포인트 가져오기

Cognitive Services 리소스를 사용하기 위해서는, 클라이언트 응용프로그램에서는 엔드포인트와 인증 키가 필요합니다.client applications need its endpoint and authentication key:

1. Azure portal에서, Cognitive Services 리소스를 선택하고 **키 및 엔트포인트** 페이지를 선택한 다음 **키1** 을 복사하여 아래의 **YOUR_COG_KEY**.를 붙여 넣는다.
2. 리소스에 있는 **엔드포인트** 를 복사해서 아래의 **YOUR_COG_ENDPOINT**.에 붙여 넣는다.
3. 셀을 선택한 다음 셀 왼쪽에있는 **셀 실행(▷)** 버튼을 클릭하여 아래 코드를 실행한다.

In []:

```
cog_key = 'YOUR_COG_KEY'
cog_endpoint = 'YOUR_COG_ENDPOINT'

print('Ready to use cognitive services at {} using key {}'.format(cog_endpoint, cog_key))
```

이제 키 및 엔드포인트를 설정했으므로 컴퓨터 비전 서비스를 사용하여 이미지를 분석할 수 있습니다.

Python에서 이 작업을 수행하려면 다음 셀을 실행하여 Azure Cognitive Services ComputerVision 패키지를 설치해야 합니다.

In []:

```
! pip install azure-cognitiveservices-vision-computervision
```

이제 준비가 다 되었습니다!

다음 셀을 실행하여 */data/vision/store_cam1.jpg* 파일에서 이미지에 대한 설명을 가져옵니다.

In []:

```
from azure.cognitiveservices.vision.computervision import ComputerVisionClient
from msrest.authentication import CognitiveServicesCredentials
from python_code import vision
import os
%matplotlib inline

# 이미지 파일 경로 가져오기
image_path = os.path.join('data', 'vision', 'store_cam1.jpg')

# Computer Vision 서비스를 위한 클라이언트 가져오기
computervision_client = ComputerVisionClient(cog_endpoint, CognitiveServicesCredentials(cog_key))

# Computer Vision 서비스로부터 설명 가져오기
image_stream = open(image_path, "rb")
description = computervision_client.describe_image_in_stream(image_stream)

# 이미지와 캡션을 보여주기Display (helper_scripts/vision.py에 있는 코드)
vision.show_image_caption(image_path, description)
```

상당히 정확한 것 같다.

다른 이미지를 만들어 보자

In []:

```
# 이미지 파일 경로 가져오기
image_path = os.path.join('data', 'vision', 'store_cam2.jpg')

# Computer Vision 서비스로부터 설명 가져오기
image_stream = open(image_path, "rb")
description = computervision_client.describe_image_in_stream(image_stream)

# 이미지와 캡션을 보여주기 Display (helper_scripts/vision.py에 있는 코드)
vision.show_image_caption(image_path, description)
```

다시 말하지만, 제안된 캡션은 꽤 정확한 것 같다..

이미지 특징 분석하기

지금까지 컴퓨터 비전 서비스를 사용하여 두 개의 이미지에 대한 설명 캡션을 생성했지만, 더 많은 작업을 수행할 수 있다. ComputerVision 서비스는 다음과 같은 세부 정보를 추출할 수 있는 분석 기능을 제공한다.:

- 이미지에서 탐지된 일반적인 개체 유형의 위치
- 이미지에서 사람의 얼굴 위치와 대략적인 연령
- 이미지에 '성인', '선정적' 또는 '잔혹물' 콘텐츠가 포함되어 있는지 여부.
- 데이터베이스의 이미지와 연결하여 쉽게 찾을 수 있는 관련 태그

다음 코드를 수행하여 쇼핑객들의 이미지를 분석해보자.

In []:

```
# 이미지 파일 경로 가져오기
image_path = os.path.join('data', 'vision', 'store_cam1.jpg')

# 분석하기 원하는 특성을 지정하기
features = ['Description', 'Tags', 'Adult', 'Objects', 'Faces']

# Computer Vision 서비스로부터 분석한 것 가져오기
image_stream = open(image_path, "rb")
analysis = computervision_client.analyze_image_in_stream(image_stream, visual_features=features)

# 분석 결과를 보여주기 S(helper_scripts/vision.py에 있는 코드 사용)
vision.show_image_analysis(image_path, analysis)
```

심화 학습 내용

이 노트북에서 살펴본 기능 외에도 Computer Vision Cognitive 서비스에는 다음과 같은 기능이 포함되어 있다.

- 이미지에서 유명인사를 식별.
- 이미지에서 브랜드 로고 감지
- 영상에서 텍스트를 읽기 위해 OCR(광학적 문자 인식)을 수행

Computer Vision Cognitive 서비스에 대한 자세한 내용은 문서를 참조하기 [Computer Vision documentation \(https://docs.microsoft.com/azure/cognitive-services/computer-vision/\)](https://docs.microsoft.com/azure/cognitive-services/computer-vision/)



✓ 600 XP ➔

Custom Vision 서비스를 사용한 이미지 분류

34분 • 모듈 • 5 단위

★★★★★ 4.8 (761)

평가하기

초급 AI 엔지니어 데이터 과학자 개발자 솔루션 아키텍처 학생

Cognitive Services

이미지 분류는 AI(인공 지능) 애플리케이션에서 일반적인 작업입니다. 기계 학습의 예측 능력을 사용하여 AI 시스템이 이미지에 기반한 실제 항목을 식별할 수 있도록 합니다.

학습 목표

Custom Vision 서비스를 사용하여 이미지 분류 솔루션을 만드는 방법에 대해 알아봅니다.

📘 책갈피 Ⓛ 컬렉션에 추가

필수 조건

Azure Portal 탐색 능력

이 모듈은 이러한 학습 경로의 일부입니다.

[Microsoft Azure에서 컴퓨터 비전 살펴보기](#)

소개

3분



Azure에서 이미지 분류 시작

3분



연습 - 이미지 분류 솔루션 만들기

25분



지식 점검

2분



요약

1분



✓ 100 XP

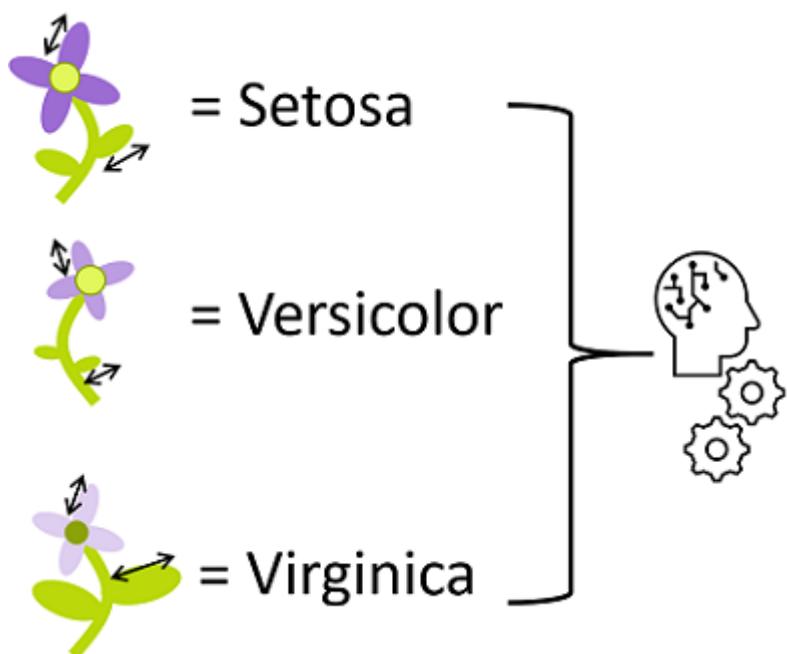


소개

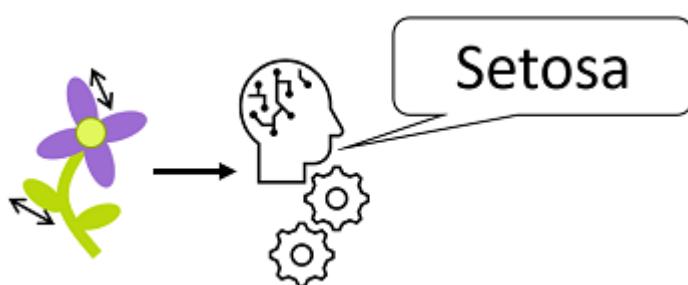
3분

'분류'는 대상이 속한 범주 또는 '클래스'를 예측하는 데 사용할 수 있는 기계 학습 기법입니다. 분류 기계 학습 모델은 '기능'이라고 부르는 입력 세트를 사용하여 가능한 각 클래스의 확률 점수를 계산하고 개체가 속할 가장 가능성성이 높은 클래스를 표시하는 '레이블'을 예측합니다.

예를 들어 꽃의 기능에는 꽃잎, 줄기, 꽃받침 및 기타 정량적 특성의 측정치가 포함될 수 있습니다. 이러한 측정치에 가장 가능성이 높은 꽃의 클래스를 계산하는 알고리즘을 적용하여 기계 학습 모델을 교육할 수 있습니다.



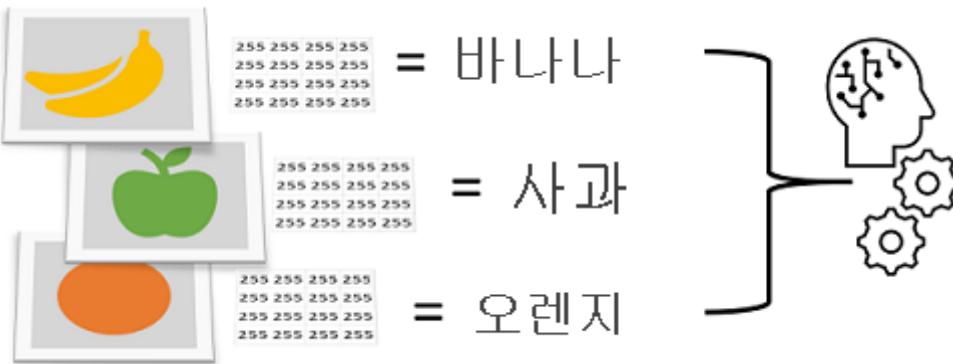
분류 모델을 만들려면 이미 알고 있는 레이블에 대한 기능으로 구성된 데이터가 필요합니다. 이 데이터를 사용하여 기능과 해당 레이블 간의 관계를 결정하도록 모델을 학습시킵니다. 모델을 학습시킨 후 새 기능 세트에 사용하여 알 수 없는 레이블 값을 예측할 수 있습니다.



이미지 분류

'이미지 분류'는 분류의 대상으로 되는 개체가 사진과 같은 이미지인 기계 학습의 한 가지 기법입니다.

모든 형태의 분류에서와 마찬가지로 이미지 분류 솔루션을 만들려면 클래스를 이미 알고 있는 기존 데이터 세트를 사용하여 기계 학습 모델을 학습시켜야 합니다. 이 경우 기존 데이터는 분류된 이미지 세트로 구성됩니다. 디지털 이미지는 픽셀 값 배열로 구성되며, 알려진 이미지 클래스를 기반으로 모델을 학습시키는 기능으로 사용됩니다.



대부분의 최신 이미지 분류 솔루션은 'CNN(나선형 인공신경망)'을 사용하여 특정 클래스에 해당하는 픽셀에서 패턴을 확인하는 '딥 러닝' 기술을 기반으로 합니다. 효과적인 CNN을 학습시키는 것은 데이터 과학 및 기계 학습에서 상당한 전문 지식이 필요한 복잡한 작업입니다. 그러나 이미지 분류 모델을 학습시키는 데 사용되는 일반적인 기술이 Microsoft Azure의 **Custom Vision** 인지 서비스에 캡슐화되었습니다. 모델을 쉽게 학습시키고 딥 러닝 기술에 대한 최소한의 지식만으로 소프트웨어 서비스로서 게시할 수 있도록 해줍니다.

이미지 분류의 사용

이미지 분류의 몇 가지 잠재적 용도는 다음과 같습니다.

- 제품 식별 - 모바일 디바이스를 사용하여 온라인이나 매장 내에서 특정 제품의 비주얼 검색 수행
- 재해 조사 - 주요 재해 대비 노력에 대한 핵심 인프라 평가. 예를 들어 항공 감시 이미지가 다리를 표시하면 다리로 분류할 수 있습니다. 다리로 분류된 모든 항목은 응급 준비 및 조사를 위해 표시할 수 있습니다.
- 의료 진단 - 엑스레이 또는 MRI 디바이스에서 이미지를 평가하면 종양으로 나타난 특정 문제 또는 의료 이미징 진단과 관련된 많은 기타 질환을 빠르게 분류할 수 있습니다.

다음 단원: Azure에서 이미지 분류 시작

계속 >

< 이전

단위 2/5 ▼

다음 >

100 XP



Azure에서 이미지 분류 시작

3분

Azure Cognitive Services 제품의 일부로 제공되는 Custom Vision 서비스를 사용하여 이미지 분류를 수행할 수 있습니다. 일반적으로 모델 학습 코드를 작성하는 것보다 쉽고 빠르게 수행할 수 있으며, 기계 학습 전문 지식이 거의 없거나 전혀 없는 사람도 효과적인 이미지 분류 솔루션을 만들 수 있습니다.

Custom Vision용 Azure 리소스

Custom Vision을 이용하여 이미지 분류 솔루션을 만드는 과정은 두 가지 주요 작업으로 이루어집니다. 먼저 기존 이미지를 사용하여 모델을 학습시킨 다음 클라이언트 애플리케이션에서 예측을 할 수 있도록 모델을 게시해야 합니다.

각 작업을 위해서는 Azure 구독의 리소스가 필요합니다. 다음과 같은 유형의 리소스를 사용할 수 있습니다.

- **Custom Vision:** Custom Vision 서비스 전용 리소스는 학습 리소스, 예측 리소스 또는 두 리소스 모두가 될 수 있습니다.
- **Cognitive Services:** 다른 많은 인지 서비스와 함께 Custom Vision을 포함하는 일반적인 Cognitive Services 리소스입니다. 이러한 유형의 리소스를 학습, 예측 또는 둘 다에 사용할 수 있습니다.

학습 리소스와 예측 리소스의 분리는 모델을 사용하여 이미지 클래스를 예측하는 클라이언트 애플리케이션과는 별도로 모델 학습의 리소스 사용률을 추적하려는 경우에 유용합니다. 그러나 이미지 분류 솔루션 개발이 약간 혼란스러워질 수 있습니다.

가장 간단한 방법은 학습과 예측에 일반 Cognitive Services 리소스를 사용하는 것입니다. 즉, 하나의 엔드포인트(서비스가 호스팅되는 HTTP 주소)와 키(클라이언트 애플리케이션에서 자체 인증 목적으로 사용하는 비밀 값)에만 신경 쓰면 됩니다.

Custom Vision 리소스를 만들도록 선택하면 '학습', '예측' 또는 '둘 다'를 선택하라는 메시지가 표시되며, "둘 다"를 선택하면 학습 및 예측을 위해 '**두 개**'의 리소스가 각각 하나씩 생성됩니다.*

또한 학습을 위해서는 전용 Custom Vision 리소스를 사용하고, 예측을 위해서는 모델을 Cognitive Services 리소스에 배포하는 혼합 및 일치 방법도 사용할 수 있습니다. 이렇게 하려면 학습 및 예측 리소스가 동일한 지역에 만들어져야 합니다.

모델 학습

분류 모델을 학습시키려면 학습 리소스에 이미지를 업로드하고 적절한 클래스 레이블로 레이블을 지정해야 합니다. 그런 다음 모델을 학습시키고 학습 결과를 평가해야 합니다.

이러한 작업을 'Custom Vision 포털'에서 수행할 수도 있고 필요한 코딩 경험이 있다면 Custom Vision 서비스 프로그래밍 언어별 SDK(소프트웨어 개발 키트) 중 하나를 사용할 수도 있습니다.*

분류를 위해 이미지를 사용할 때 고려해야 할 주요 사항 중 하나는 해당 객체에 대한 이미지가 충분한지 확인하는 것이며, 그러한 이미지는 다양한 각도에서 본 객체에 대한 것이어야 합니다.

모델 평가

모델 학습 프로세스는 Custom Vision 서비스가 데이터 중 일부를 사용하여 모델을 반복적으로 학습시키는 반복적 프로세스이지만, 모델을 평가하기 위해 일부를 보류합니다. 학습 프로세스가 끝날 때 학습된 모델의 성능은 다음과 같은 평가 메트릭에 의해 표시됩니다.

- **정밀도**: 모델에서 올바르게 이루어진 클래스 예측의 비율은 얼마인가요? 예를 들어 모델에서 10개의 이미지가 오렌지로 예측되고 8개의 이미지가 실제로 오렌지인 경우 정밀도는 0.8(80%)입니다.
- **재현율**: 모델에서 올바르게 식별된 클래스 예측의 비율은 얼마인가요? 예를 들어 사과 이미지 10개가 있고 모델에서 그중 7개의 이미지를 찾은 경우 재현율은 0.7(70%)입니다.
- **AP(평균 정밀도)**: 정밀도와 재현율을 모두 고려한 전체 메트릭입니다.

예측을 위한 모델 사용

모델을 학습시키고 평가된 성능에 만족하면 모델을 예측 리소스에 게시할 수 있습니다. 모델을 게시할 때 이름을 지정할 수 있습니다(기본값은 X가 모델을 학습한 횟수인 "반복 X"입니다).

모델을 사용하려면 클라이언트 애플리케이션 개발자는 다음 정보가 필요합니다.

- **프로젝트 ID**: 모델을 학습시키기 위해 만든 Custom Vision 프로젝트의 고유 ID입니다.
- **모델 이름**: 게시하는 동안 모델에 할당된 이름입니다.
- **예측 엔드포인트**: 모델을 게시한 '예측' 리소스(학습 리소스가 '아님')에 대한 엔드포인트의 HTTP 주소입니다.*
- **_ 예측 키***: 모델을 게시한 '예측' 리소스(학습 리소스가 *****'아님')의 인증 키입니다.

다음 단원: 연습 - 이미지 분류 솔루션 만들기

계속 >

< 이전

단위 3/5 ▼

다음 >

✓ 100 XP



연습 - 이미지 분류 솔루션 만들기

25분

Custom Vision 서비스를 사용한 이미지 분류에 관해 알아보는 가장 좋은 방법은 스스로 탐색하는 것입니다.

시작하기 전에

이 연습을 완료하려면 다음이 필요합니다.

- Microsoft Azure 구독. 구독이 아직 없다면 <https://azure.microsoft.com/free>에서 평가판에 가입할 수 있습니다.
- MicrosoftDocs/ai-fundamentals GitHub 리포지토리에서 Jupyter Notebook을 실행할 수 있는 Python 환경. GitHub의 랩 파일을 자신의 Python 환경(있는 경우)에 복제하거나, 아래 지침에 따라 Azure 구독에서 Azure Machine Learning 작업 영역을 만들 수 있습니다.

ⓘ 참고

이 모듈은 Azure Machine Learning 작업 영역을 사용하는 여러 가지 모듈 중 하나입니다. **Azure AI 기본 사항** 인증의 준비 과정에서 이 모듈을 완료하는 경우, 작업 영역을 한 번 만들고 해당 인증 트랙의 모든 모듈에 다시 사용할 수 있습니다. 연습을 완료한 후에는 정리 지침에 따라 모듈 간 컴퓨팅 리소스를 중지해야 합니다.

Azure Machine Learning 작업 영역 만들기

Azure 구독에 Azure Machine Learning 작업 영역이 이미 있는 경우 새 브라우저 탭에서 [Azure Machine Learning Studio](#)로 이동하고 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다. 없는 경우 다음 단계에 따라 새 작업 영역을 만듭니다.

1. Azure 구독과 연결된 Microsoft 계정을 사용하여 [Azure Portal](#)에 로그인합니다.
2. + 리소스 만들기를 선택하고, *Machine Learning*을 검색한 뒤 다음 설정을 사용하여 새 *Machine Learning* 리소스를 만듭니다.
 - **작업 영역 이름:** 원하는 고유한 이름을 입력
 - **구독:** 자신의 Azure 구독
 - **리소스 그룹:** 고유한 이름의 새 리소스 그룹 만들기
 - **위치:** 사용 가능한 위치 선택

3. 작업 영역 리소스가 만들어질 때까지 기다립니다(몇 분이 걸릴 수 있음). 그런 다음 포털로 이동하고 작업 영역의 **개요** 페이지에서 Azure Machine Learning Studio를 실행(또는 새 브라우저 탭을 열고 <https://ml.azure.com> 으로 이동)한 다음 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다.
4. Azure Machine Learning Studio에서 왼쪽 상단에 있는 **☰** 아이콘을 설정/해제하면 인터페이스에 다양한 페이지가 표시됩니다. 해당 페이지를 사용하여 작업 영역에서 리소스를 관리할 수 있습니다.

컴퓨팅 인스턴스 만들기

이 연습에서 사용된 Notebook을 실행하려면 Azure Machine Learning 작업 영역에 컴퓨팅 인스턴스가 필요합니다. 컴퓨팅 인스턴스가 이미 있는 경우 시작하고, 없는 경우 다음 지침에 따라 새로 만듭니다.

1. [Azure Machine Learning Studio](#)에서 **관리** 아래에 있는 **컴퓨팅** 페이지를 확인합니다.
2. **컴퓨팅 인스턴스** 탭에서 다음 설정을 사용하여 새 컴퓨팅 인스턴스를 만듭니다.
 - **지역**: 사용 가능한 지역 선택
 - **가상 머신 유형**: CPU
 - **가상 머신 크기**: Standard_DS11_v2
 - **컴퓨팅 이름**: 고유한 이름 입력
3. 컴퓨팅 인스턴스가 시작될 때까지 기다립니다(1분 정도 걸릴 수 있음).

연습 파일 다운로드

이 모듈(및 기타 관련 모듈)에 사용되는 파일은 GitHub 리포지토리에 게시되는데, 이를 자신의 Python 환경에 복제해야 합니다. 이전 모듈에서 **ai-fundamentals** 리포지토리를 아직 복제하지 않은 경우 다음 단계를 사용하여 Azure Machine Learning 작업 영역에 복제합니다.

1. [Azure Machine Learning Studio](#)에서 **Notebooks** 페이지(**작성자** 아래)를 확인합니다. 이 페이지에는 Notebooks 실행에 사용할 수 있는 Notebook 편집기가 포함되어 있습니다.
2. **내 파일**에서 **□** 단추를 사용하여 다음 설정에 따라 새 파일을 만듭니다.
 - **파일 위치**: `Users/your user name`
 - **파일 이름**: Get-Files
 - **파일 형식**: Notebook
 - **이미 있는 경우 덮어쓰기**: 선택됨
3. 새 Notebook이 만들어지면 이전에 만든 컴퓨팅 인스턴스가 **컴퓨팅** 상자에서 선택되었는지, 그리고 상태가 **실행 중**인지 확인합니다. 그런 다음 Notebook에서 만든 사각형 셀에 다음 코드를 붙여넣습니다.

```
!git clone https://github.com/MicrosoftDocs/ai-fundamentals
```

4. 셀 옆에 있는 ▶ 단추를 사용하여 포함된 코드를 실행합니다. 이렇게 하면 GitHub에서 연습 파일이 복제됩니다.
5. 코드의 실행이 끝나고 파일의 체크아웃이 완료되면 내 파일 아래에 있는 ⌂ 단추를 사용하여 폴더 보기를 새로 고치고, **ai-fundamentals**라는 이름의 폴더가 만들어졌는지 확인합니다. 이 폴더에는 연습에 사용된 Notebook 및 기타 파일이 포함되어 있습니다.
6. Get-Files.ipynb Notebook 탭을 닫습니다.

연습 완료

Python 환경을 설정하고 **ai-fundamentals** 리포지토리를 복제했으면 Custom Vision에서 이미지 분석을 살펴볼 준비가 된 것입니다.

1. **ai-fundamentals** 폴더에서 **Image Classification.ipynb** Notebook을 엽니다. Azure Machine Learning Studio에서 Notebook 편집기를 사용하는 경우, « 단추로 파일 탐색기 창을 축소하여 Notebook 탭에 집중할 수 있는 공간을 좀 더 마련합니다.
2. Notebook의 정보를 읽고 포함된 코드 셀을 순서대로 실행합니다.

정리

Azure Machine Learning Studio에서 컴퓨팅 인스턴스를 사용한 경우 Azure 크레딧을 불필요하게 사용하지 않도록 이를 중지해야 합니다.

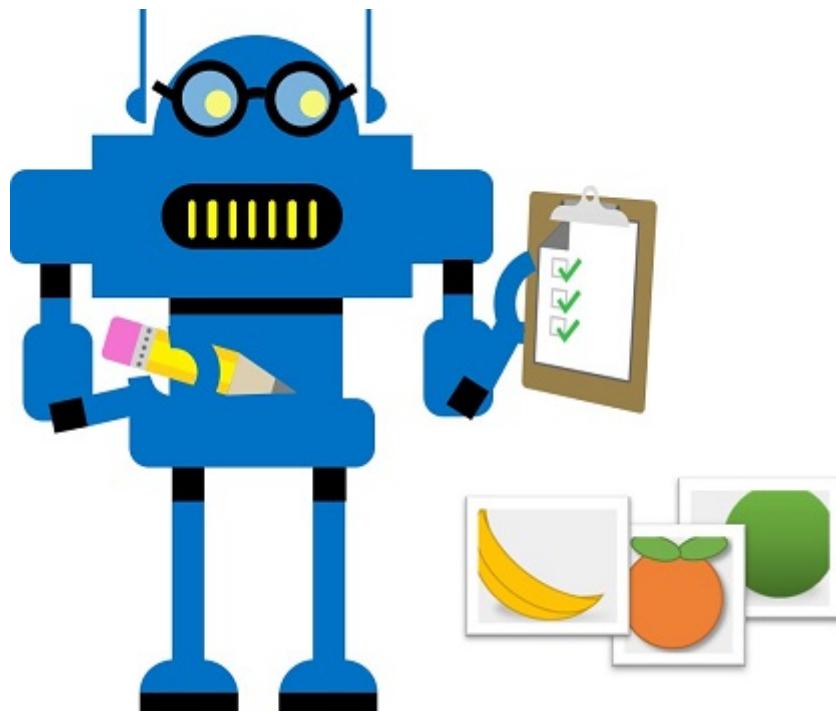
1. Azure Machine Learning Studio에서 관리 아래에 있는 컴퓨팅 페이지를 확인합니다.
2. 컴퓨팅 인스턴스 탭에서 컴퓨팅 인스턴스를 선택하고 중지 단추를 사용하여 중지합니다.

다음 단원: 지식 점검

계속 >

이미지 분류하기(Classification)

Computer Vision Cognitive 서비스는 이미지를 작업할 때 유용하게 사전 제작된 모델을 제공하지만 종종 컴퓨터 비전을 위한 자신만의 모델을 교육해야 한다. 예를 들어, Northwind Traders 소매 회사가 계산대에서 카메라로 찍은 이미지를 기반으로 고객이 구매하고자 하는 식료품 품목을 식별하는 자동 계산 시스템을 생성하려고 한다고 가정해 보자. 이렇게 하려면 이미지를 분류하여 구입하려는 항목을 식별할 수 있는 분류 모델을 교육해야 한다.



Azure에서 **Custom Vision** Cognitive 서비스를 사용하여 이미 존재하는 이미를 기초로하여 이미지 분류모델을 학습시킬 수 있다. 이미지 분류 모델을 생성하기 위해서는 두 가지 요소가 있다. 첫번째로 이미 존재하는 이미지를 사용하여 각기 다른 클래스로 인지하도록 모델을 학습시켜야 한다. 그리고 나서 모델이 학습하였다면 응용프로그램이 이용하도록 서비스를 게시해야 한다.

Custom Vision 리소스 만들기

Custom Vision 서비스를 사용하려면 모델을 *training* 하는 데 사용할 수 있는 Azure 리소스와 애플리케이션을 위해 모델을 게시할 수 있는 리소스가 필요하다. 두 작업 중 하나 또는 둘 다에 대한 리소스는 일반적인 **Cognitive** 서비스 리소스 또는 특별한 **CustomVision** 리소스가 될 수 있다. 이러한 각 작업에 대해 동일한 Cognitive 서비스 리소스를 사용하거나, 각 작업에 대해 서로 다른 리소스(동일한 지역)를 사용하여 비용을 별도로 관리할 수 있다.

새로운 **Custom Vision** 리소스를 만들기 위해서는 다음과 같은 순서로 수행하면 된다.

1. 새 브라우저 탭에서 <https://portal.azure.com> (<https://portal.azure.com>)에서 Azure 포털을 열고 Azure 구독과 연결된 Microsoft 계정을 사용하여 로그인한다.
2. + 리소스 만들기 버튼을 선택하고, *Custom Vision*을 검색한 뒤에 다음과 같이 **Custom Vision** 리소스를 생성한다:
 - **만들기 옵션:** 둘 다
 - **구독:** Azure 구독 옵션
 - **리소스 그룹:** 유일한 이름으로 리소스 그룹 생성하기
 - **이름:** 유일한 이름(영문과 숫자 적용)
 - **학습 위치:** 가능한 위치 선택

- 교육 가격 책정 계층: F0
- 예측 위치: 학습위치와 동일한 지역 선택
- 예측 가격 책정위치: F0

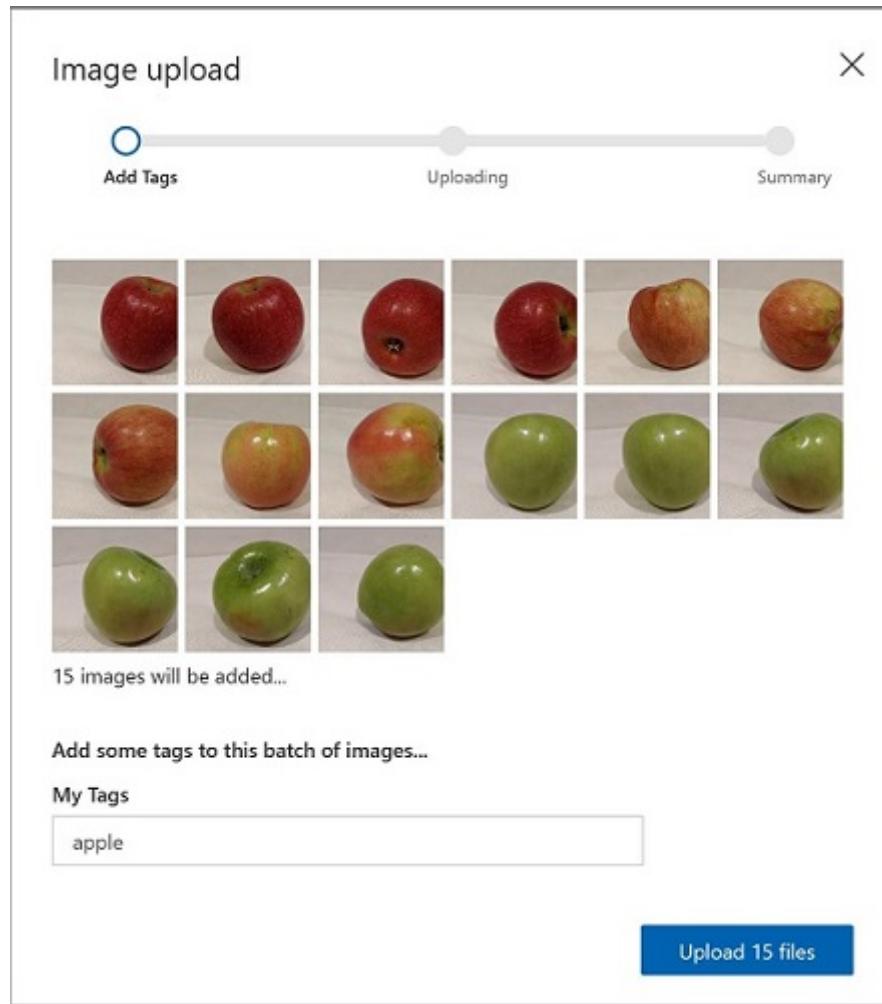
알림: 여러분들이 이미 F0 Custom Vision 서비스를 사용하고 있다면 여기서는 **S0**을 선택함.

3. 리소스가 생성되길 기다리고 두개의 Custom Vision 리소스가 생성되어야 하는데 하나는 학습을 위해서이고 하나는 예측을 위해서이다. 이런 것들은 여러분이 생성한 리소스 그룹에 종속되어 보여진다.

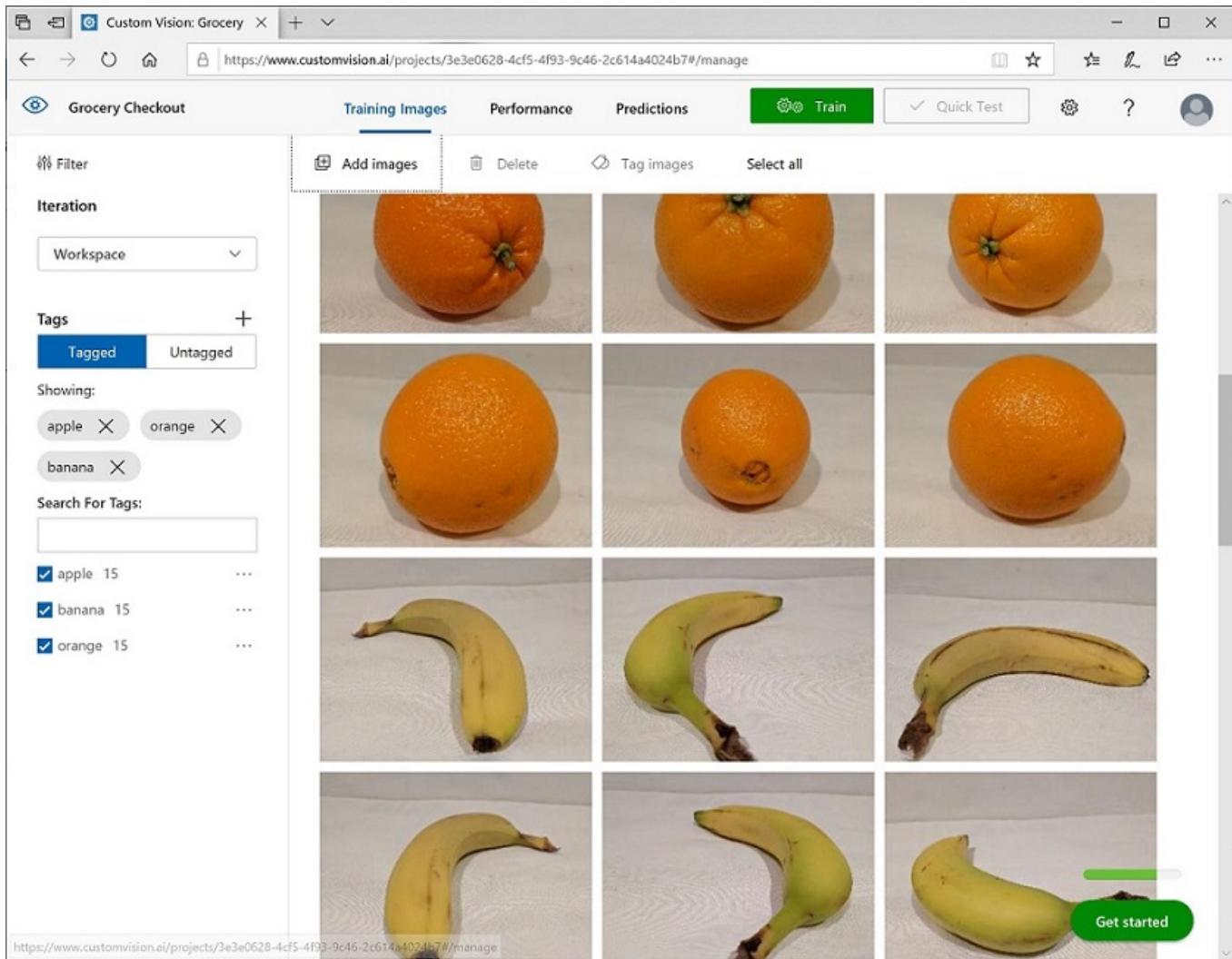
Custom Vision 프로젝트 만들기

객체 감지 모델을 학습시키기 위해 학습 리소스를 바탕으로 하여 Custom Vision 객체를 생성하여야 한다. 이렇게 하기 위해서는 Custom Vision 포털을 사용하게 될 것이다.

1. 학습 이미지 파일을 다운로드하여 압축을 해제한다. (<https://aka.ms/fruit-images>).
2. 브라우저의 다른 탭을 열고, Custom Vision portal 사이트 <https://customvision.ai> (<https://customvision.ai>)를 입력한다. 입력창에 Azure 구독에 사용된 Microsoft 계정 정보를 입력하고 서비스 계약서에 동의한다..
3. Custom Vision 포털에서, 다음과 같은 내용으로 새로운 프로젝트를 입력한다.
 - **Name:** Grocery Checkout
 - **Description:** 채소에 대한 이미지 분류
 - **Resource:** 앞에서 만든 Custom Vision 리소스
 - **Project Types:** Classification
 - **Classification Types:** Multiclass (single tag per image)
 - **Domains:** Food
4. **[+] Add images**를 클릭하고 앞에서 압축을 해제한 **apple** 폴더에 있는 모든 파일을 선택한다. 이미지를 업로드하고 나서 이미지에 **apple**로 태그를 단다.



1. 앞에서와 같은 순서로 **banana** 폴더의 이미지에 태그를 *banana*로 달고 , **orange** 폴더에 있는 이미지들에는 *orange*라는 태그를 부여한다.
2. Custom Vision 프로젝트 내에 업로드한 이미지들을 확인해본다-각 분야마다 15개의 이미지들이 존재한다. 예를들면 다음과 같다.



- Custom Vision 프로젝트에서 이미지를 위에 있는 **Train**을 클릭하여 태그가 된 이미지를 이용하여 분류 모델을 학습시킨다. **Quick Training** 옵션을 클릭하고 학습 순서가 끝나기를 기다린다(이 작업은 1분 이상 걸릴 수 있다).
- 모델이 학습되었다면 *Precision*, *Recall*, 그리고 *AP* 성능 Metrics를 확인한다 - 이런 값들은 분류 모델의 정확도를 예측하는데 사용되며 모두 높게 나와야 좋다.

모델을 테스트하기

응용 프로그램에 사용하기 위해 게시하는 작업 전에 모델을 테스트해보아야 한다.

- 성능 Metrics위에 있는 **Quick Test**를 클릭한다.
- Image URL** 상자에서, `https://aka.ms/apple-image` 라 입력하고 **→** 를 클릭한다.
- 모델에 의해 예측된 결과를 확인한다-아래와 같이 *apple*에 대한 Probability 점수가 가장 높아야 한다:

Quick Test X



Image URL
ka.ms/apple-image →
or
Browse local files

File formats accepted: jpg, png, bmp
File size should not exceed: 4mb

Using model trained in
Iteration
Iteration 1

Predictions

Tag	Probability
apple	99.9%
orange	0%
banana	0%

1. Quick Test 창을 닫는다.

이미지 분류 모델을 게시하고 사용하기

이제 학습시킨 모델을 게시하고 클라이언트 응용프로그램에서 사용할 수 있게 되었다.

1. 다음과 같은 내용을 입력하고 ✓ Publish를 클릭하여 학습시킨 모델을 게시한다.

- **Model name:** groceries
- **Prediction Resource:** 앞에서 생성한 예측 리소스.

2. 게시후에 프로젝트의 설정을 확인하기 위해 **Performance**페이지의 왼쪽 오른편에 있는 *settings* (⚙) 아이콘을 클릭한다. 그런 후 (왼쪽에 있는)**General**에서, **Project Id**를 복사하여 아래의 코드셀에 붙여 넣는다 (**YOUR_PROJECT_ID**를 대체한다).

Project Settings

General

Project Name*: Grocery Checkout

Project Id: 3e3e0628-4cf5-4f93-9c46-2c614a4024b7#

Description: Image classification for groceries

Usage: ①

- 45 training images uploaded; 4955 remain
- 3 tags created; 47 remain
- 1 iterations saved; 9 remain

Domains:

- General
- Food
- Landmarks
- Retail
- General (compact)
- Food (compact)

Resources:

qmalc-cv
Subscription:
Resource Group:
Resource Kind: Custom Vision Training

Key: [Redacted]

Endpoint: [Redacted]

Resource Id: [Redacted]

Pricing Tier: F0

Change Pricing Tier

1 projects created; 1 remain

Note: 이 학습이 초기에 **Custom Vision** 리소스를 사용하는 대신 **Cognitive Services** 리소스를 사용했다면, 프로젝트 설정의 오른편에 있는 키와 엔트포인트를 복사해서 아래의 코드셀에 붙여 넣고 결과를 확인하기 위해서 실행한다. 그렇지 않으면, Custom Vision 예측 리소스에 대한 키와 엔드포인트를 얻기 위해 아래와 같은 작업을 완료한다.

1. A **Project Settings** 설정 페이지의 위쪽 왼편에 있는 **Projects Gallery** (👁) 아이콘을 클릭하여 Custom Vision 포털 홈페이지로 이동하면 프로젝트 목록이 나타난다.
2. Custom Vision 포털 홈페이지에서 윗쪽 오른편에서 **settings** (⚙) 아이콘을 클릭하여 Custom Vision 서비스에 대한 설정을 본다. 그리고 나서 **Resources** 밑에 **prediction** 리소스를 클릭하고 resource (학습리소스가 아님) 키와 엔드포인트값을 복사해서 아래의 **YOUR_KEY** 와 **YOUR_ENDPOINT**를 대체한다.

Resources: [create new](#)

- qmalccv-Prediction

Subscription:
Resource Group:
Resource Kind: Custom Vision Prediction

Key: c5265fa2e2c2486eb3664afc8d25580f

Endpoint: https://qmalccv-prediction.cognitiveservices.azure.com/

Resource Id:
- Pricing Tier: F0

[Change Pricing Tier](#)

0 predictions made; 10000 remain until reset
- qmalccv

Subscription:
Resource Group:
Resource Kind: Custom Vision Training

1. 프로젝트 ID, 키, 엔드포인트값 변수 값을 설정한 후에 **Run cell (▷)** 버튼(셀의 왼편에 있음)을 클릭하여 시작합니다

In []:

```
project_id = 'YOUR_PROJECT_ID'
cv_key = 'YOUR_KEY'
cv_endpoint = 'YOUR_ENDPOINT'

model_name = 'groceries' # 이 모델 이름은 모델 게시작업때 설정한 이름과 대소문자까지 일치해야 한다!
print('Ready to predict using model {} in project {}'.format(model_name, project_id))
```

Python에서 Custom Vision 서비스를 사용하기 위해서는 Azure Cognitive Services Custom Vision 패키지를 설치해야 한다.

In []:

```
!pip install azure-cognitiveservices-vision-customvision
```

이제 Custom Vision 클라이언트가 키와 앤드포인트를 사용하여 만든 Custom Vision 분류 모델에 연결할 수 있다.

게시한 모델을 사용하여 테스트 이미지를 잘 분석하는지 알아보기 위해 아래와 같은 코드를 실행한다.

Note: 코드의 너무 자세한 내용에 대해서 걱정하지 말라. 이건 Computer Vision SDK for Python를 사용하여 /data/image-classification/test-fruit폴더에 있는 각 이미지에 대한 분류 예측을 위한 것이다.

In []:

```
from azure.cognitiveservices.vision.customvision.prediction import CustomVisionPredictionClient
from msrest.authentication import ApiKeyCredentials
import matplotlib.pyplot as plt
from PIL import Image
import os
%matplotlib inline

# data/vision/test 폴더에 있는 이미지들을 가져온다.
test_folder = os.path.join('data', 'image-classification', 'test-fruit')
test_images = os.listdir(test_folder)

# 예측 서비스의 인스턴스를 생성한다.
credentials = ApiKeyCredentials(in_headers={"Prediction-key": cv_key})
custom_vision_client = CustomVisionPredictionClient(endpoint=cv_endpoint, credentials=credentials)

# 결과를 화면에 표시하도록 그림을 생성한다.
fig = plt.figure(figsize=(16, 8))

# 이미지를 가져와서 각 이미지에 대한 예측 분류값을 보여준다.
print('Classifying images in {} ...'.format(test_folder))
for i in range(len(test_images)):
    # 이미지를 알고 Custom Vision 모듈을 사용하여 분류하도록 준비한다.
    image_contents = open(os.path.join(test_folder, test_images[i]), "rb")
    classification = custom_vision_client.classify_image(project_id, model_name, image_contents.read())
    # 예측 값은 각 이미지 태그에 대한 예측값을 포함하고 확률을 내림차순으로 정리해서 보여준다.
    prediction = classification.predictions[0].tag_name
    # 예측 클래스가 포함된 이미지를 나타낸다.
    img = Image.open(os.path.join(test_folder, test_images[i]))
    a=fig.add_subplot(len(test_images)/3, 3,i+1)
    a.axis('off')
    imgplot = plt.imshow(img)
    a.set_title(prediction)
plt.show()
```

다행히 여러분의 이미지 분류 모델은 정확하게 이미지 있는 채소를 바르게 찾아 낸다.

추가 학습

Custom Vision 서비스는 이 학습에서 본 것들 보다 너 능률적인 것으로 보여준다. 예를 들면 Custom Vision 서비스를 사용하여 객체 감지 모델을 생성하는데 Custom Vision 모델을 사용할 수 있다(이것은 이미지 내에 있는 객체들을 구분할 뿐만 아니라 이미지에 포함되어 있는 개체의 위치를 보여주는 튀어오르는 상자를 구분할 수 있다).

Custom Vision cognitive 서비스에 대해서 더 많은 것을 알고 싶다면, [Custom Vision documentation](https://docs.microsoft.com/azure/cognitive-services/custom-vision-service/home) (<https://docs.microsoft.com/azure/cognitive-services/custom-vision-service/home>)를 참조하면 된다.



✓ 600 XP ➔

Custom Vision 서비스를 사용하여 이미지의 개체 감지

39분 • 모듈 • 5 단위

★★★★★ 4.8 (699)

평가하기

초급 AI 엔지니어 데이터 과학자 개발자 솔루션 아키텍처 학생

Cognitive Services

개체 감지는 AI(인공 지능) 에이전트가 이미지 또는 카메라 피드에서 특정 유형의 개체를 식별하고 찾을 수 있는 Computer Vision의 한 형태입니다.

학습 목표

Custom Vision 서비스를 사용하여 개체 감지 솔루션을 만드는 방법을 알아봅니다.

📘 책갈피 Ⓛ 컬렉션에 추가

필수 조건

Azure Portal 탐색 능력

이 모듈은 이러한 학습 경로의 일부입니다.

[Microsoft Azure에서 컴퓨터 비전 살펴보기](#)

소개

3분



Azure에서 개체 감지 시작

3분



연습 - 개체 감지 솔루션 만들기

30분



지식 점검

2분



요약

1분



✓ 100 XP

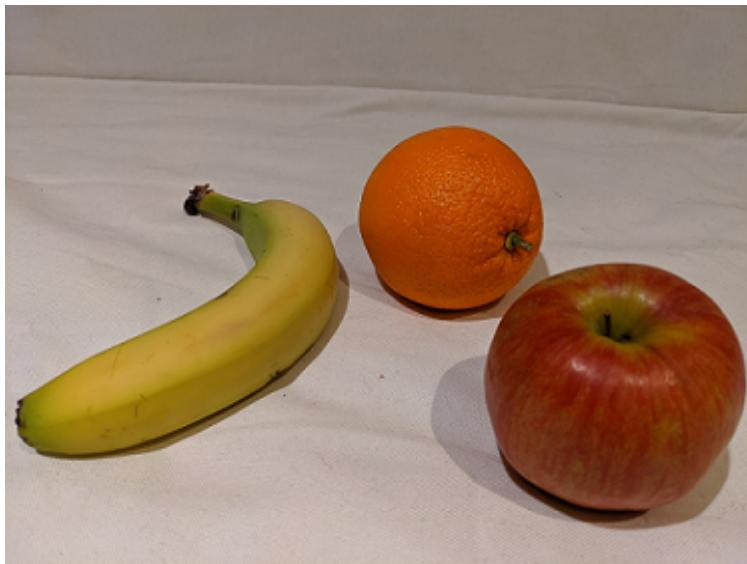


소개

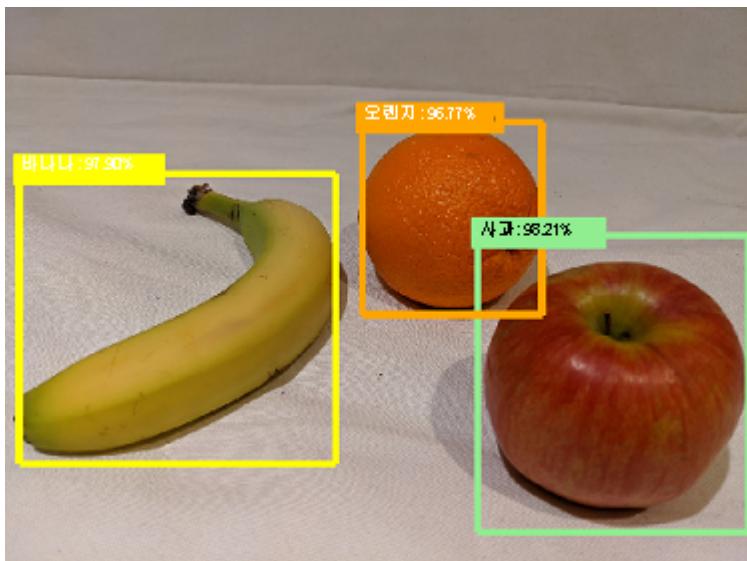
3분

개체 감지는 기계 학습 기반의 Computer Vision의 한 형태로, 이미지에서 개체의 개별 유형을 인식한 다음 해당 개체의 위치를 식별하도록 모델을 학습시킵니다.

예를 들어 다음과 같은 이미지를 생각해 보세요.



개체 감지 모델을 사용하여 이 이미지의 개별 개체를 식별하고 다음 정보를 반환할 수 있습니다.



개체 감지 모델은 다음 정보를 반환합니다.

- 이미지에서 식별된 각 개체의 클래스.
- 개체 분류의 확률 점수(예측된 클래스가 정확할 신뢰도로 해석할 수 있음).

- 각 개체의 경계 상자 좌표.

① 참고

개체 검색과 이미지 분류 비교

이미지 분류는 모델에 포함된 주요 주제를 기준으로 이미지를 분류하도록 모델을 학습시키는 기계 학습 기반 Computer Vision 형태입니다. 개체 감지는 여기서 더 나아가 이미지 내의 개별 개체를 분류하고 개체의 위치를 나타내는 경계 상자의 좌표를 반환합니다.

개체 감지의 사용

개체 감지의 몇 가지 샘플 애플리케이션은 다음과 같습니다.

- 소화기 또는 기타 비상 장비를 찾아 건물의 안전을 평가.
- 자율 주행 차량 또는 차선 보조 기능이 있는 차량을 위한 소프트웨어 만들기.
- 의료 진단을 위해 알려진 개체를 감지할 수 있는 MRI 또는 X선 등의 의료 이미징.

다음 단원: Azure에서 개체 감지 시작

계속 >

[〈 이전](#)

단위 2/5 ▼

[다음 〉](#)

✓ 100 XP



Azure에서 개체 감지 시작

3분

고급 딥 러닝 기술을 사용하여 개체 감지 기계 학습 모델을 만들 수 있습니다. 하지만 이 방법에는 상당한 전문 지식과 많은 양의 학습 데이터가 필요합니다. Azure의 **Custom Vision** 인지 서비스를 사용하면 최소한의 딥 러닝 전문 지식과 더 적은 학습 이미지를 사용하여 여러 Computer Vision 시나리오의 요구 사항을 충족하는 개체 감지 모델을 만들 수 있습니다.

Custom Vision용 Azure 리소스

Custom Vision을 사용하여 개체 감지 솔루션을 만드는 과정은 다음 세 가지 주요 작업으로 이루어집니다. 먼저 이미지를 업로드하고 태그를 지정해야 모델을 학습시킬 수 있습니다. 마지막으로 클라이언트 애플리케이션이 예측 생성에 사용할 수 있도록 모델을 게시해야 합니다.

각 작업을 위해서는 Azure 구독의 리소스가 필요합니다. 다음과 같은 유형의 리소스를 사용할 수 있습니다.

- **Custom Vision** : Custom Vision 서비스 전용 리소스는 학습 리소스 또는 예측 리소스가 될 수 있습니다.
- **Cognitive Services** : 다른 많은 인지 서비스와 함께 Custom Vision을 포함하는 일반적인 Cognitive Services 리소스입니다. 이러한 유형의 리소스를 학습, 예측 또는 둘 다에 사용할 수 있습니다.

학습 리소스와 예측 리소스의 분리는 모델을 사용하여 이미지 클래스를 예측하는 클라이언트 애플리케이션과는 별도로 모델 학습의 리소스 사용률을 추적하려는 경우에 유용합니다. 그러나 이미지 분류 솔루션 개발이 약간 혼란스러워질 수 있습니다.

가장 간단한 방법은 학습과 예측에 일반 Cognitive Services 리소스를 사용하는 것입니다. 즉, 하나의 엔드포인트(서비스가 호스팅되는 HTTP 주소)와 키(클라이언트 애플리케이션에서 자체 인증 목적으로 사용하는 비밀 값)에만 신경 쓰면 됩니다.

Custom Vision 리소스를 만들도록 선택하면 ‘학습’, ‘예측’ 또는 ‘둘 다’를 선택하라는 메시지가 표시되며, “둘 다”를 선택하면 학습 및 예측을 위해 ‘**두 개**’의 리소스가 각각 하나씩 생성됩니다.*

또한 학습을 위해서는 전용 Custom Vision 리소스를 사용하고, 예측을 위해서는 모델을 Cognitive Services 리소스에 배포하는 혼합 및 일치 방법도 사용할 수 있습니다. 이렇게 하려면 학습 및 예측 리소스가 동일한 지역에 만들어져야 합니다.

이미지 태그 지정

개체 감지 모델을 학습시키려면 먼저 학습 이미지 집합의 클래스 및 경계 상자 좌표에 태그를 지정해야 합니다. 이 프로세스는 시간이 많이 걸릴 수 있지만 'Custom Vision 포털'은 이를 간단하게 할 수 있는 그래픽 인터페이스를 제공합니다.* 인터페이스는 불연속 개체가 감지되는 이미지 영역을 자동으로 제안하며, 이러한 제안된 경계 상자에 클래스 레이블을 적용하거나 끌어서 경계 상자 영역을 조정할 수 있습니다. 또한 초기 데이터 세트를 사용하여 태그를 지정하고 학습시킨 후 Computer Vision 서비스는 스마트 태그 지정을 사용하여 학습 데이터 세트에 추가하는 이미지의 클래스와 경계 상자를 제안할 수 있습니다.

개체 감지를 위한 학습 이미지에 태그를 지정할 때 중요한 고려 사항은 해당 개체의 이미지가 충분 한지 확인하는 것입니다. 가능하면 여러 각도의 이미지가 있는 것이 좋고 경계 상자는 각 개체 주위에 딱 붙게 정의되는 것이 좋습니다.

모델 학습 및 평가

모델을 학습시키려면 Custom Vision 포털을 사용할 수도 있고 필요한 코딩 경험이 있다면 Custom Vision 서비스 프로그래밍 언어별 SDK(소프트웨어 개발 키트) 중 하나를 사용할 수도 있습니다. 개체 감지 모델 학습은 각 이미지 내의 학습 이미지, 클래스, 개체 수에 따라 다소 시간이 걸릴 수 있습니다.

모델 학습 프로세스는 Custom Vision 서비스가 데이터 중 일부를 사용하여 모델을 반복적으로 학습시키는 반복적 프로세스이지만 일부는 모델 평가를 위해 보류합니다. 학습 프로세스가 끝날 때 학습된 모델의 성능은 다음과 같은 평가 메트릭에 의해 표시됩니다.

- **정밀도** : 모델에서 올바르게 식별된 클래스 예측의 비율은 얼마인가요? 예를 들어 모델에서 10개의 이미지가 오렌지로 예측되고 8개의 이미지가 실제로 오렌지인 경우 정밀도는 0.8(80%)입니다.
- **재현율** : 모델에서 올바르게 이루어진 클래스 예측의 비율은 얼마인가요? 예를 들어 사과 이미지 10개가 있고 모델이 그중 7개의 이미지를 찾은 경우 재현율은 0.7(70%)입니다.
- **평균 정밀도(mAP)** 모든 클래스에서의 정밀도와 재현율을 모두 고려한 전체 메트릭입니다.

예측을 위한 모델 사용

모델을 학습시키고 평가된 성능에 만족하면 모델을 예측 리소스에 게시할 수 있습니다. 모델을 게시할 때 이름을 지정할 수 있습니다(기본값은 X가 모델을 학습한 횟수인 "반복 X"입니다).

모델을 사용하려면 클라이언트 애플리케이션 개발자에게 다음 정보가 필요합니다.

- **프로젝트 ID** : 모델을 학습시키기 위해 만든 Custom Vision 프로젝트의 고유 ID입니다.
- **모델 이름** : 게시하는 동안 모델에 할당된 이름입니다.

- **예측 엔드포인트** : 모델을 게시한 '예측' 리소스(학습 리소스가 '아님')에 대한 엔드포인트의 HTTP 주소입니다.*_
 - _ 예측 키*: 모델을 게시한 '예측' 리소스(학습 리소스가 **** **'아님')의 인증 키입니다.
-

다음 단원: 연습 - 개체 감지 솔루션 만들기

계속 >

< 이전

단위 3/5 ▼

다음 >

100 XP



연습 - 개체 감지 솔루션 만들기

30분

Custom Vision 서비스를 사용한 개체 감지에 대해 알아보는 가장 좋은 방법은 스스로 탐색하는 것입니다.

시작하기 전에

이 연습을 완료하려면 다음이 필요합니다.

- Microsoft Azure 구독. 구독이 아직 없다면 <https://azure.microsoft.com/free>에서 평가판에 가입할 수 있습니다.
- MicrosoftDocs/ai-fundamentals GitHub 리포지토리에서 Jupyter Notebook을 실행할 수 있는 Python 환경. GitHub의 랩 파일을 자신의 Python 환경(있는 경우)에 복제하거나, 아래 지침에 따라 Azure 구독에서 Azure Machine Learning 작업 영역을 만들 수 있습니다.

ⓘ 참고

이 모듈은 Azure Machine Learning 작업 영역을 사용하는 여러 가지 모듈 중 하나입니다.

Azure AI 기본 사항 인증의 준비 과정에서 이 모듈을 완료하는 경우, 작업 영역을 한 번 만들고 해당 인증 트랙의 모든 모듈에 다시 사용할 수 있습니다. 연습을 완료한 후에는 정리 지침에 따라 모듈 간 컴퓨팅 리소스를 중지해야 합니다.

Azure Machine Learning 작업 영역 만들기

Azure 구독에 Azure Machine Learning 작업 영역이 이미 있는 경우 새 브라우저 탭에서 [Azure Machine Learning Studio](#)로 이동하고 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다. 없는 경우 다음 단계에 따라 새 작업 영역을 만듭니다.

1. Azure 구독과 연결된 Microsoft 계정을 사용하여 [Azure Portal](#)에 로그인합니다.
2. + 리소스 만들기를 선택하고, *Machine Learning*을 검색한 뒤 다음 설정을 사용하여 새 *Machine Learning* 리소스를 만듭니다.
 - **작업 영역 이름:** 원하는 고유한 이름을 입력
 - **구독:** 자신의 Azure 구독
 - **리소스 그룹:** 고유한 이름의 새 리소스 그룹 만들기
 - **위치:** 사용 가능한 위치 선택

3. 작업 영역 리소스가 만들어질 때까지 기다립니다(몇 분이 걸릴 수 있음). 그런 다음 포털로 이동하고 작업 영역의 **개요** 페이지에서 Azure Machine Learning Studio를 실행(또는 새 브라우저 탭을 열고 <https://ml.azure.com> 으로 이동)한 다음 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다.
4. Azure Machine Learning Studio에서 왼쪽 상단에 있는 **☰** 아이콘을 설정/해제하면 인터페이스에 다양한 페이지가 표시됩니다. 해당 페이지를 사용하여 작업 영역에서 리소스를 관리할 수 있습니다.

컴퓨팅 인스턴스 만들기

이 연습에서 사용된 Notebook을 실행하려면 Azure Machine Learning 작업 영역에 컴퓨팅 인스턴스가 필요합니다. 컴퓨팅 인스턴스가 이미 있는 경우 시작하고, 없는 경우 다음 지침에 따라 새로 만듭니다.

1. [Azure Machine Learning Studio](#)에서 **관리** 아래에 있는 **컴퓨팅** 페이지를 확인합니다.
2. **컴퓨팅 인스턴스** 탭에서 다음 설정을 사용하여 새 컴퓨팅 인스턴스를 만듭니다.
 - **지역**: 사용 가능한 지역 선택
 - **가상 머신 유형**: CPU
 - **가상 머신 크기**: Standard_DS11_v2
 - **컴퓨팅 이름**: 고유한 이름 입력
3. 컴퓨팅 인스턴스가 시작될 때까지 기다립니다(1분 정도 걸릴 수 있음).

연습 파일 다운로드

이 모듈(및 기타 관련 모듈)에 사용되는 파일은 GitHub 리포지토리에 게시되는데, 이를 자신의 Python 환경에 복제해야 합니다. 이전 모듈에서 **ai-fundamentals** 리포지토리를 아직 복제하지 않은 경우 다음 단계를 사용하여 Azure Machine Learning 작업 영역에 복제합니다.

1. [Azure Machine Learning Studio](#)에서 **Notebooks** 페이지(**작성자** 아래)를 확인합니다. 이 페이지에는 Notebooks 실행에 사용할 수 있는 Notebook 편집기가 포함되어 있습니다.
2. **내 파일**에서 **□** 단추를 사용하여 다음 설정에 따라 새 파일을 만듭니다.
 - **파일 위치**: `Users/your user name`
 - **파일 이름**: Get-Files
 - **파일 형식**: Notebook
 - **이미 있는 경우 덮어쓰기**: 선택됨
3. 새 Notebook이 만들어지면 이전에 만든 컴퓨팅 인스턴스가 **컴퓨팅** 상자에서 선택되었는지, 그리고 상태가 **실행 중**인지 확인합니다. 그런 다음 Notebook에서 만든 사각형 셀에 다음 코드를 붙여넣습니다.

```
!git clone https://github.com/MicrosoftDocs/ai-fundamentals
```

4. 셀 옆에 있는 ▶ 단추를 사용하여 포함된 코드를 실행합니다. 이렇게 하면 GitHub에서 연습 파일이 복제됩니다.
5. 코드의 실행이 끝나고 파일의 체크아웃이 완료되면 내 파일 아래에 있는 ⌂ 단추를 사용하여 폴더 보기를 새로 고치고, **ai-fundamentals**라는 이름의 폴더가 만들어졌는지 확인합니다. 이 폴더에는 연습에 사용된 Notebook 및 기타 파일이 포함되어 있습니다.
6. Get-Files.ipynb Notebook 탭을 닫습니다.

연습 완료

Python 환경을 설정하고 **ai-fundamentals** 리포지토리를 복제했으면 Custom Vision에서 개체 감지를 살펴볼 준비가 된 것입니다.

1. **ai-fundamentals** 폴더에서 **Object Detection.ipynb** Notebook을 엽니다. Azure Machine Learning Studio에서 Notebook 편집기를 사용하는 경우, <> 단추로 파일 탐색기 창을 축소하여 Notebook 탭에 집중할 수 있는 공간을 좀 더 마련합니다.
2. Notebook의 정보를 읽고 포함된 코드 셀을 순서대로 실행합니다.

정리

Azure Machine Learning Studio에서 컴퓨팅 인스턴스를 사용한 경우 Azure 크레딧을 불필요하게 사용하지 않도록 이를 중지해야 합니다.

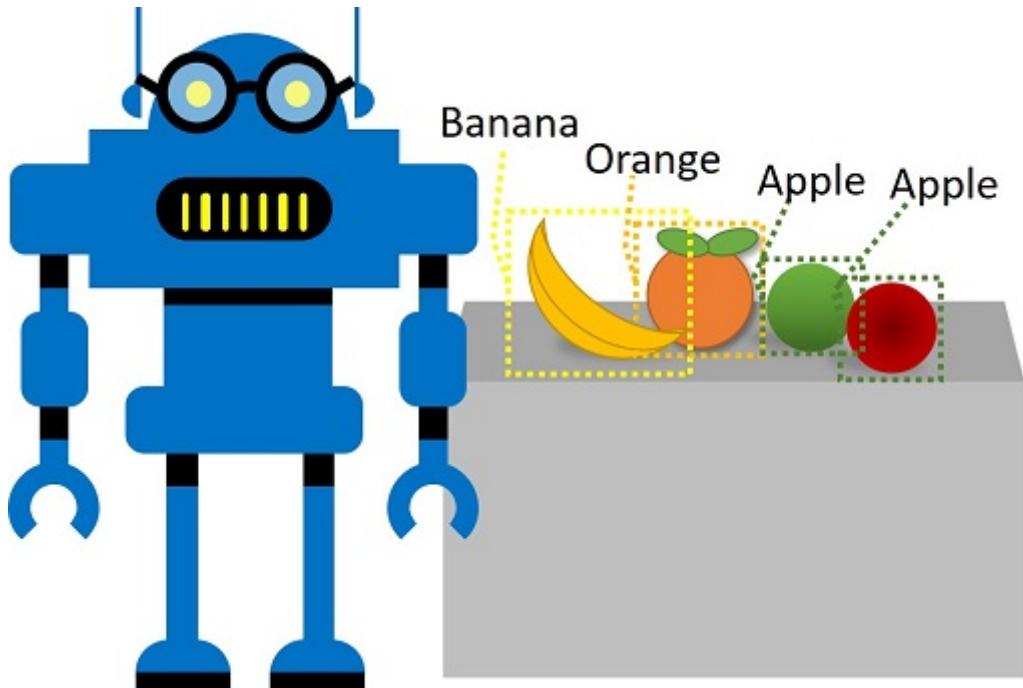
1. Azure Machine Learning Studio에서 관리 아래에 있는 컴퓨팅 페이지를 확인합니다.
2. 컴퓨팅 인스턴스 탭에서 컴퓨팅 인스턴스를 선택하고 중지 단추를 사용하여 중지합니다.

다음 단원: 지식 점검

계속 >

개체 감지

개체 감지는 기계 학습 모델을 학습하여 이미지에서 사물의 개별 인스턴스를 분류하고 그 위치를 표시하는 표시 상자를 표시하는 컴퓨터 비전이다. 이것을 이미지 분류("이것이 이미지란 무엇인가?"라는 질문에 모델이 답하는)에서 모델에게 "이 이미지에 어떤 물체가 있고, 어디에 있는가?"라고 물어볼 수 있는 솔루션 구축으로 발전했다고 생각할 수 있다.



예를 들어, 식료품점에서 카메라를 사용하여 컨베이어 벨트를 스캔하는 자동 체크 시스템을 구현하기 위해 객체 감지 모델을 사용할 수 있으며, 벨트에 각 품목을 배치하고 개별적으로 스캔할 필요 없이 특정 품목을 식별할 수 있다.

Microsoft Azure에 있는 **Custom Vision** Cognitive 서비스는 클라우드에서 자체의 개체 감지 모델을 생성하고 게시하는 기능을 제공한다..

Custom Vision 리소스 생성하기

Custom Vision 서비스를 사용하기 위해서 모델을 학습하는데 사용할 수 있는 Azure 리소스가 필요하고 응용 프로그램이 사용할 수 있도록 게시할 수 있는 리소스가 필요하다. 이런 각각의 작업에 같은 리소스를 이용할 수도 있고 또는 리소스에 대한 비용을 따로 처리하기 위해 같은 리전에서 리소스를 분리해서 생성할 수 있다. 각 작업(또는 두 작업 모두)을 위한 리소스는 일반적인 **Cognitive** 서비스리소스가 될 수도 있고 특별한 **Custom Vision** 리소스가 될 수 있다. 다음 순서로 새로운 **Custom Vision**리소스를 만들수 있다(만일 이미 존재하는 리소스가 있다면 그것을 활용할 수도 있다).

1. 새 브라우저 탭에서 <https://portal.azure.com>에서 (<https://portal.azure.com>에서) Azure 포털을 열고 Azure 구독과 연결된 Microsoft 계정을 사용하여 로그인한다.
2. +를 선택한다. 리소스 만들기 버튼을 선택하고,, **Custom Vision**을 검색한 뒤에 다음과 같이 Custom Vision** 리소스를 생성한다:
3. + 리소스 만들기 버튼을 선택하고, **Custom Vision**을 검색한 뒤에 다음과 같이 **Custom Vision** 리소스를 생성한다:
 - **만들기 옵션:** 둘 다
 - **구독:** Azure 구독 옵션
 - **리소스 그룹:** 유일한 이름으로 리소스 그룹 생성하기
 - **이름:** 유일한 이름(영문과 숫자 적용)

- **학습 위치:** 가능한 위치 선택
- **교육 가격 책정 계층:** F0
- **예측 위치:** 학습위치와 동일한 지역 선택
- **예측 가격 책정위치:** F0

알림: 여러분들이 이미 F0 Custom Vision서비스를 사용하고 있다면 여기서는 **S0**을 선택함.

1. 리소스가 생성되길 기다린다.

Custom Vision 프로젝트 생성하기

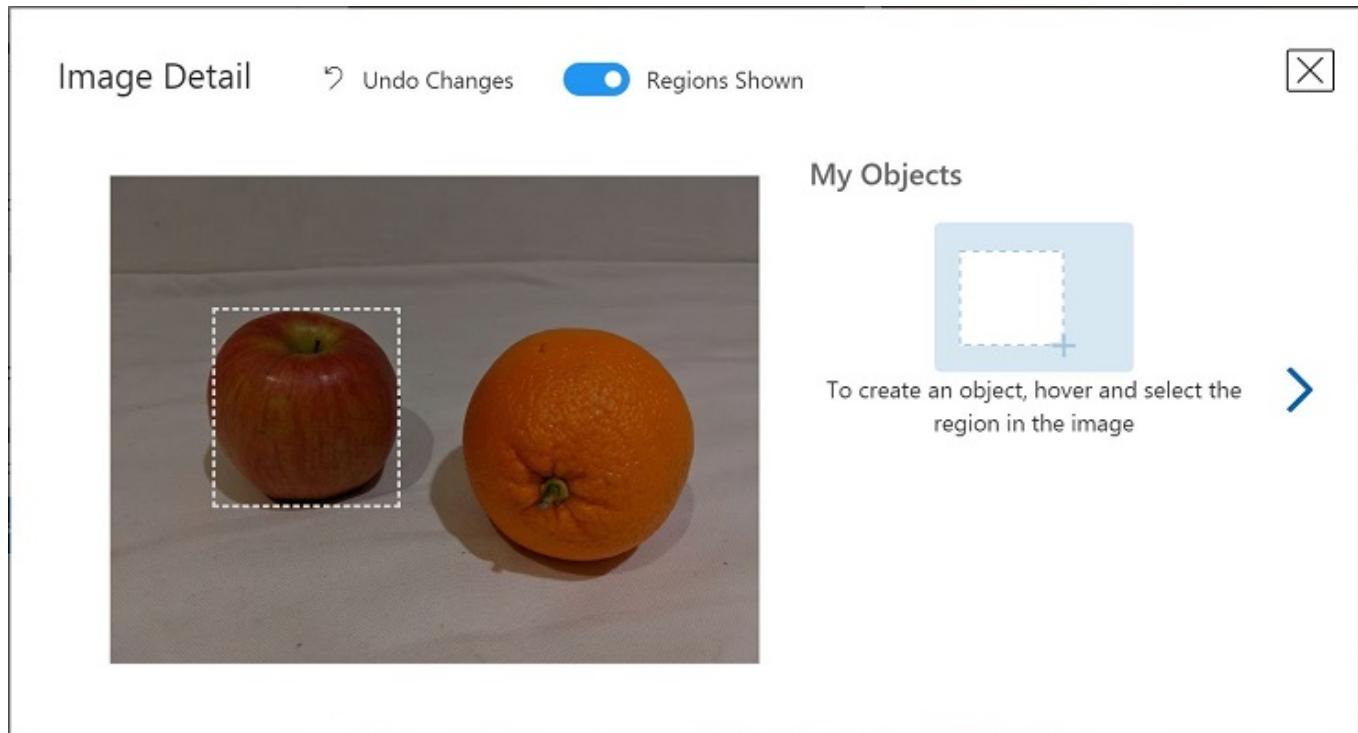
객체 감지 모델을 학습시키기 위해 학습 리소스를 바탕으로 하여 Custom Vision 프로젝트를 생성하여야 한다. 이렇게 하기 위해서는 Custom Vision 포털을 사용하게 될 것이다.

1. 브라우저의 다름 탭을 열고, Custom Vision portal 사이트 <https://customvision.ai> (<https://customvision.ai>)를 입력한다. 입력창에 Azure 구독에 사용된 Microsoft 계정 정보를 입력한다..
2. 새로운 프로젝트를 생성할 때 다음과 같이 입력하다.:
 - **Name:** Grocery Detection
 - **Description:** 채소에 대한 이미지 감지.
 - **Resource:** 앞에서 만든 Custom Vision 리소스
 - **Project Types:** Object Detection
 - **Domains:** General
3. 프로젝트가 만들어지고 브라우저에서 열릴때까지 기다린다.

이미지 추가하고 태그하기

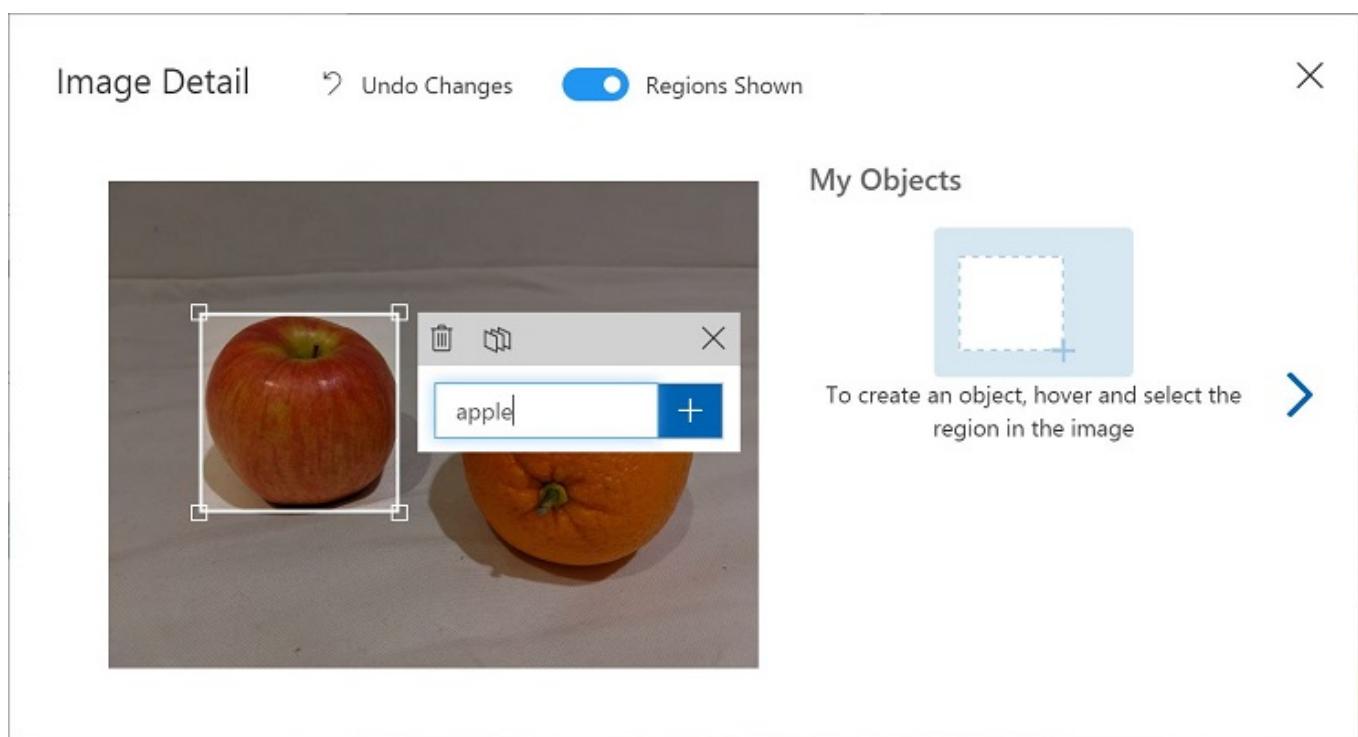
개체 검지 모델을 학습시키기 위해서는 모델이 인지하도록 하기 원하는 클래스들이 포함되어 있는 이미지를 업로드해야 하고 이미지에 있는 각 개체에 대해서 표시 상자를 이용하여 태그를 달아야 한다.

1. 학습 이미지를 <https://aka.ms/fruit-objects> (<https://aka.ms/fruit-objects>)에서 다운로드하고 압축을 해제한다. 압축해제된 폴더에는 과일 이미지들이 포함되어 있다.
2. Custom Vision 포털에서, 개체 감지 프로젝트를 선택하고 **Add images** 를 선택한다음 압축해제된 이미지를 업로드한다.
3. 이미지들이 업로드가 된 후에 첫번째 이미지를 선택하여 연다.
4. 이미지에 있는 각 개체위에 마우스를 올려두고 아래와 같이 자동으로 검지된 영역을 표시할 때까지 기다린다. 그 다음 해당 개체를 서낵하고 개체를 둘러 쌀 수 있도록 크기를 조절한다.

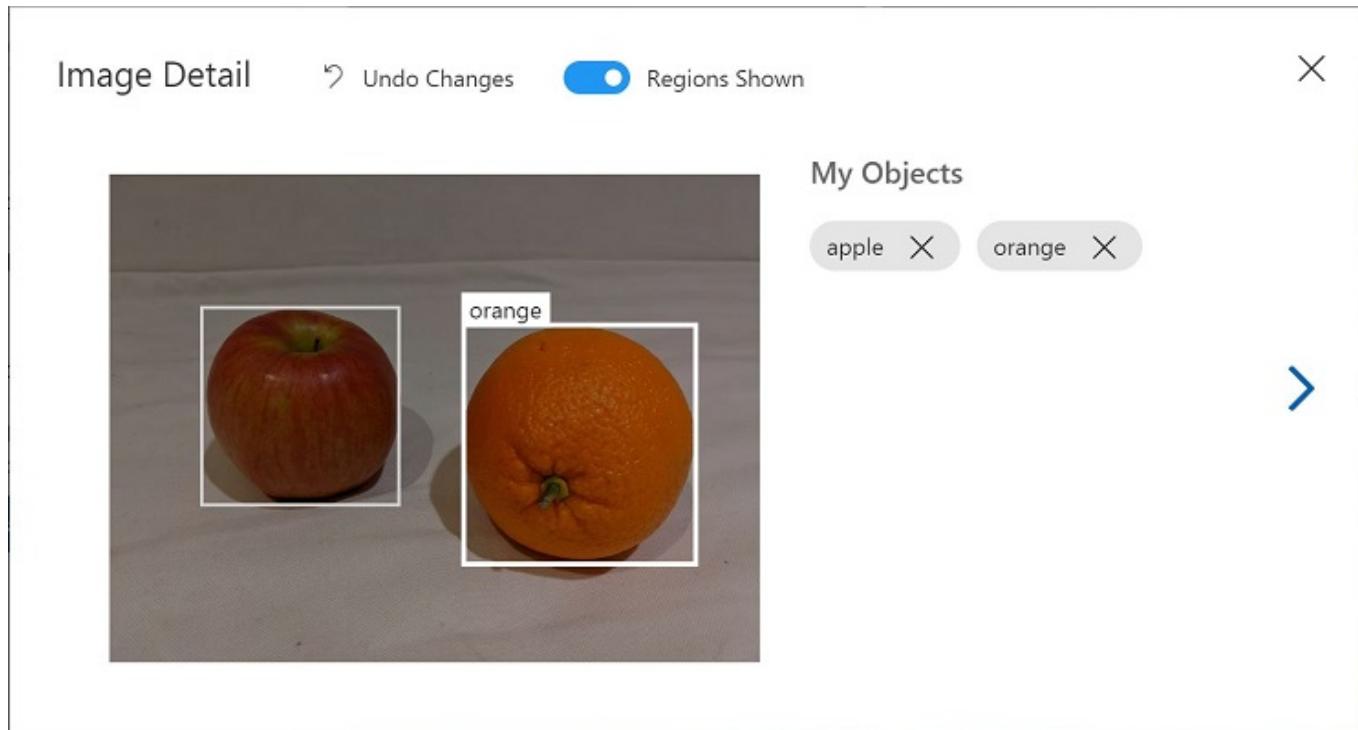


선택적으로 영역을 정하기 위해 개체 주위로 마우스를 드래그할 수 있다.

1. 개체 주위로 감싼 후에 개체의 종류에 따라 새로운 태그를 추가한다(아래의 예시처럼 *apple*, *banana*, *orange* 블인다).



1. 이미지에 다른 개체에 대해서도 선택하고 태그를 하고 영역을 다시 조절하거나 필요에 따라 새로운 태그를 부여한다.



1. > 를 사용하여 다음 오른쪽 이미지로 넘어가서 개체에 태그를 부여한다. 그리고 나서 전체 이미지들에 대해서 apple, banana, 그리고 orange로 태그를 부여한다.
2. 마지막 이미지에 대해서 태그 작업을 마쳤다면 **Image Detail** 편집기를 닫고 **Training Images** 페이지에서, **Tags** 아래, **Tagged** 를 선택하여 태그한 이미지들을 모두 본다.

The screenshot shows the 'Training Images' tab of the Custom Vision Detect Produce interface. It displays a grid of six images of various fruits (apples, oranges, and bananas) used for training. On the left sidebar, there are filters for 'Workspace' and 'Tags'. Under 'Tags', 'Tagged' is selected, showing 18 tagged images. Below this, there is a search bar for tags and a list of available tags: 'apple 18', 'banana 20', and 'orange 20'. At the bottom right of the main area, there is a green 'Get started' button.

모델 학습하기 및 테스트하기

이제 프로젝트에 있는 이미지들에 태그 작업을 마쳤으므로 모델을 학습시킬 차례다.

- Custom Vision 프로젝트에서 **Train**을 클릭하여 태그된 이미지를 이용하여 개체 검지 모델 학습시킨다. **Quick Training** 옵션을 클릭한다.
- 학습이 끝나길 기다린다(보통 10분 정도 걸릴 것이다). 그리고 나서 *Precision*, *Recall*, 및 *mAP* 성능 매트릭을 확인한다-이 값들은 분류의 예측 정확도를 측정하는데 사용되며 높을수록 좋다.
- 페이지의 오른쪽 상단에 있는 **Quick Test**를 클릭하고, **Image URL** 상자에 <https://aka.ms/apple-orange> 을 입력하고 생성하는 예측값을 확인한다. **Quick Test**창을 닫는다.

개체 검지 모델 게시하고 사용하기

이제 학습시킨 모델을 게시하고 클라이언트 응용프로그램에서 사용할 수 있게 되었다.

- 페이지의 상단 왼편에 있는 ✓ **Publish** 를 클릭하여 학습한 모델을 다음과 같은 내용으로 게시를 한다.
 - Model name:** detect-produce
 - Prediction Resource:** Custom Vision의 **prediction** 리소스.
- 게시가 끝나면 **Performance**페이지의 상단 오른편의 *settings* (⚙) 아이콘을 클릭하여 프로젝트의 설정을 확인한다. 그리고 나서 **General** (왼쪽)아래에 있는 내용 중 **Project Id** 를 복사하여 아래 코드 셀의 **YOUR_PROJECT_ID** 란에 붙여넣기를 한다.

_Note: 이 학습이 초기에 Custom Vision 리소스를 사용하는 대신 Cognitive Services 리소스를 사용했다면, 프로젝트 설정의 오른편에 있는 키와 엔드포인트를 복사해서 아래의 코드 셀에 붙여 넣고 결과를 확인하기 위해서 실행한다. 그렇지 않으면, Custom Vision 예측 리소스에 대한 키와 엔드포인트를 얻기 위해 아래와 같은 작업을 완료한다.

- Project Settings** 페이지의 상단 왼편에 있는 *Projects Gallery* (👁) 아이콘을 클릭하여 Custom Vision 포털 홈페이지로 돌아가서 프로젝트 목록들이 나타나도록 한다.
- Custom Vision 포털 홈페이지에서 윗쪽 오른편에서 *settings* (⚙) 아이콘을 클릭하여 Custom Vision 서비스에 대한 설정을 본다. 그리고 나서 **Resources** 밑에 **prediction** 리소스를 클릭하고 **resource** (학습리소스가 아님) 키와 **엔드포인트**값을 복사해서 아래의 **YOUR_KEY** 와 **YOUR_ENDPOINT**를 대체한다..
- 프로젝트 ID, 키, 엔드포인트값 변수 값을 설정한 후에 **Run cell** (▷) 버튼(셀의 왼편에 있음)을 클릭하여 실행합니다.

In []:

```
project_id = 'YOUR_PROJECT_ID' # 프로젝트 ID로 바꿈
cv_key = 'YOUR_KEY' # 예측 리소스의 첫번째 키로 바꿈
cv_endpoint = 'YOUR_ENDPOINT' # 예측리소스의 엔드포인트로 바꿈

model_name = 'detect-produce' # 이것은 모델을 게시할 때 사용했던 이름과 정확히 동일하게 맞추어야 함(대소문자 포함)!
print('Ready to predict using model {} in project {}'.format(model_name, project_id))
```

Python에서 Custom Vision 서비스를 사용하기 위해 Azure Cognitive Services Custom Vision 패키지를 설치해야 한다.

In []:

```
!pip install azure-cognitiveservices-vision-customvision
```

이제 여러분의 키와 앤드포인트를 사용하여 Custom Vision 클라이드로 Custom Vision 개체 감지 모델에 연결할 수 있다. 아래의 코드 셀을 실행하여 이미지에 있는 개별 개체 항목을 검지하는데 모델을 사용할 수 있다.

Note: 코드의 너무 자세한 내용에 대해서 걱정하지 말라. 이것은 Computer Vision SDK for Python를 사용하여 이미지를 모델로 보내고 검지된 개체에 대한 예측값을 가져오는데 사용된다. 각 예측값에는 클래스 이름(*apple*, *banana*, 또는 *orange*)과 이미지에서 예측된 개체가 검지된 부분을 표시 상자로 나타낸다. 이 정보를 이용하여 이미지에 있는 각 개체를 둘러싼 레이블 상자를 그리게 된다.

In []:

```

from azure.cognitiveservices.vision.customvision.prediction import CustomVisionPredictionClient
from msrest.authentication import ApiKeyCredentials
from matplotlib import pyplot as plt
from PIL import Image, ImageDraw, ImageFont
import numpy as np
import os
%matplotlib inline

# 테스트 이미지를 불러오고 Dimension을 저장한다.
test_img_file = os.path.join('data', 'object-detection', 'produce.jpg')
test_img = Image.open(test_img_file)
test_img_h, test_img_w, test_img_ch = np.array(test_img).shape

# 개체 검지 모델을 사용하기 위한 예측 클라이언트를 가져온다.
credentials = ApiKeyCredentials(in_headers={"Prediction-key": cv_key})
predictor = CustomVisionPredictionClient(endpoint=cv_endpoint, credentials=credentials)

print('Detecting objects in {} using model {} in project {}...'.format(test_img_file, model_name, project_id))

# 테스트 이미지에 있는 개체를 검지한다.
with open(test_img_file, mode="rb") as test_data:
    results = predictor.detect_image(project_id, model_name, test_data)

# 결과를 나타내기 위한 그림을 그린다.
fig = plt.figure(figsize=(8, 8))
plt.axis('off')

# 검지된 개체 주위로 상자를 이미지에 나타낸다.
draw = ImageDraw.Draw(test_img)
lineWidth = int(np.array(test_img).shape[1]/100)
object_colors = {
    "apple": "lightgreen",
    "banana": "yellow",
    "orange": "orange"
}
for prediction in results.predictions:
    color = 'white' # 기본값으로 'other' 개체 태그를 부여한다.
    if (prediction.probability*100) > 50:
        if prediction.tag_name in object_colors:
            color = object_colors[prediction.tag_name]
        left = prediction.bounding_box.left * test_img_w
        top = prediction.bounding_box.top * test_img_h
        height = prediction.bounding_box.height * test_img_h
        width = prediction.bounding_box.width * test_img_w
        points = ((left,top), (left+width,top), (left+width,top+height), (left,top+height), (left,top))
        draw.line(points, fill=color, width=lineWidth)
        plt.annotate(prediction.tag_name + ": {:.2f}%".format(prediction.probability * 100),(left,top), backgroundcolor=color)
plt.imshow(test_img)

```

각 예측에 대한 검지된 개체와 확률등을 나타내는 예측 결과의 내용을 확인한다.



✓ 600 XP ➔

Face 서비스를 통해 얼굴 감지 및 분석

24분 • 모듈 • 5 단위

★★★★★ 4.8 (667)

평가하기

초급 AI 엔지니어 데이터 과학자 개발자 솔루션 아키텍처 학생

Cognitive Services

얼굴 감지, 분석 및 인식은 AI(인공 지능) 솔루션의 중요한 기능입니다. Azure의 Face 인식 서비스를 사용하면 이러한 기능을 애플리케이션에 쉽게 통합할 수 있습니다.

학습 목표

Face 인식 서비스를 사용하여 이미지에서 얼굴을 감지 및 분석하는 방법을 알아봅니다.

📘 책갈피 Ⓛ 컬렉션에 추가

필수 조건

Azure Portal 탐색 능력

이 모듈은 이러한 학습 경로의 일부입니다.

[Microsoft Azure에서 컴퓨터 비전 살펴보기](#)

소개

3분



Azure의 Face 분석 시작하기

3분



연습 - Face 서비스를 통해 얼굴 감지 및 분석

15분



지식 점검

2분



요약

1분



✓ 100 XP



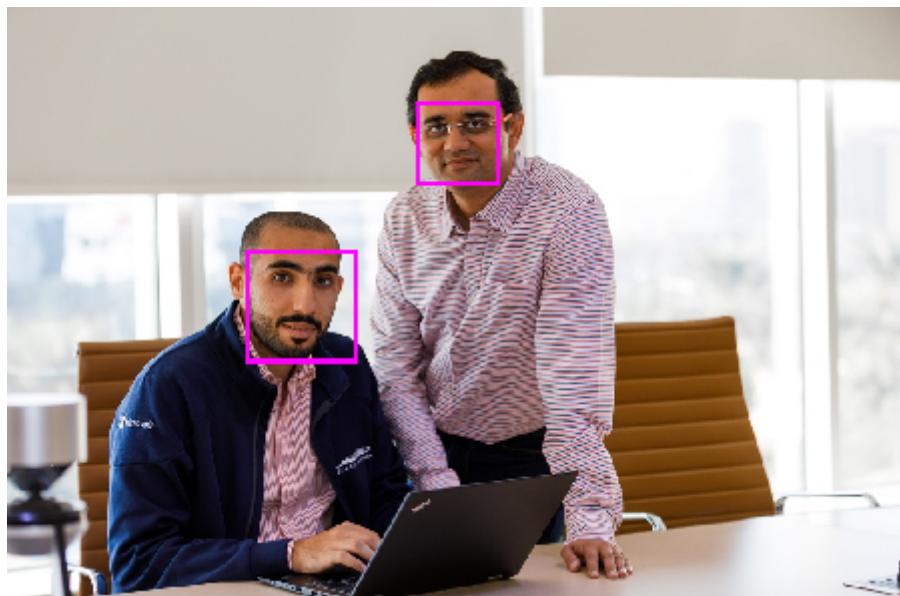
소개

3분

얼굴 감지 및 분석은 알고리즘을 사용하여 이미지 또는 비디오 콘텐츠에서 사람의 얼굴을 찾고 분석하는 AI(인공 지능) 영역 중 하나입니다.

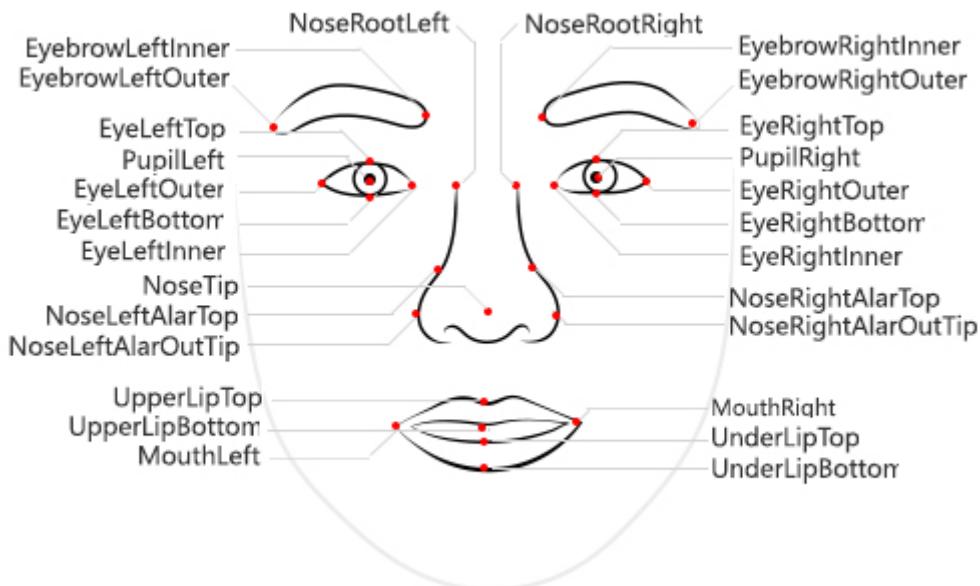
얼굴 감지

얼굴 감지에는 다음과 같이 일반적으로 얼굴 주위에 사각형을 형성하는 경계 상자 좌표를 반환하여 사람의 얼굴을 포함하는 이미지 영역을 식별하는 작업이 포함됩니다.



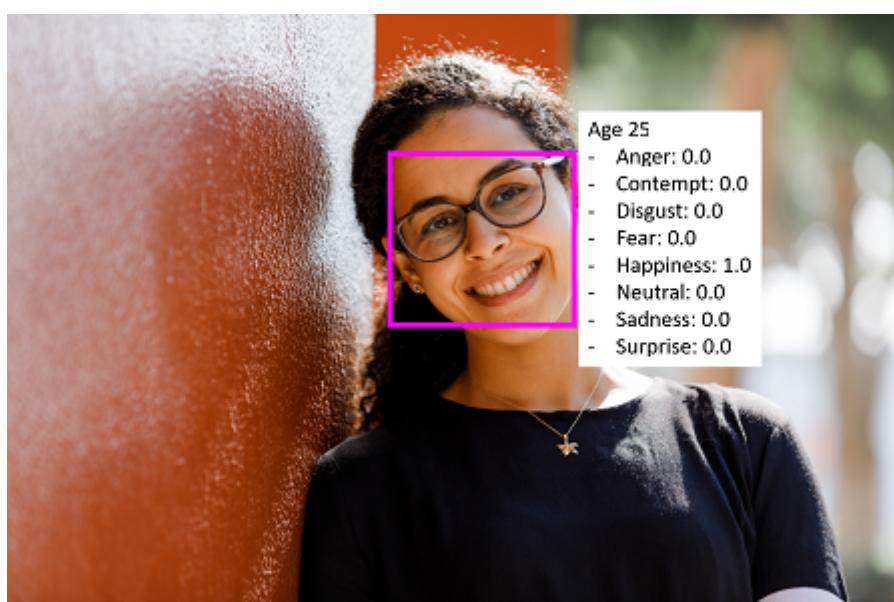
얼굴 분석

간단한 얼굴 감지를 넘어, 일부 알고리즘은 얼굴 랜드마크(코, 눈, 눈썹, 입술 등)와 같은 다른 정보를 반환할 수 있습니다.



Copyright (c) Microsoft. All rights reserved.

이러한 얼굴 랜드마크는 다음과 같이 연령 또는 감정 상태 등 인적 정보를 추론하는 기계 학습 모델을 학습시키는 기능으로서 사용할 수 있습니다.



얼굴 인식

얼굴 분석의 또 한 가지 응용 분야는 얼굴 특징에서 알려진 개인을 식별하도록 기계 학습 모델을 학습시키는 것입니다. 이 용도는 보다 일반적으로 얼굴 인식으로 알려져 있으며, 인식하려는 각 개인의 여러 이미지를 사용하여 학습되지 않은 새로운 이미지에서 해당 개인을 감지할 수 있도록 모델을 학습시키는 것입니다.



얼굴 감지 및 분석의 용도

얼굴 감지, 분석 및 인식은 다양한 응용 분야에서 활용됩니다. 예를 들면 다음과 같습니다.

- 보안 - 얼굴 인식은 보안 애플리케이션을 빌드하는 데 사용할 수 있으며, 스마트폰 운영 체제에서 디바이스 잠금을 해제하기 위한 용도로 점점 더 보급화되고 있습니다.
- 소셜 미디어 - 얼굴 인식을 사용하여 사진에서 알려진 친구를 자동으로 태그 지정할 수 있습니다.
- 지능형 모니터링 - 운전자가 도로를 보고 있는지, 모바일 디바이스를 보고 있는지 아니면 피로의 징후를 보이는지 알아보기 위해 운전자의 얼굴을 모니터링하는 시스템이 자동차에 포함될 수 있습니다.
- 광고 - 이미지에서 얼굴을 분석하여 적절한 인구 통계학적 대상 그룹에 광고를 직접 전달 할 수 있습니다.
- 실종자 - 공공 카메라 시스템과 얼굴 인식을 사용하여 실종자가 이미지 프레임에 있는지 식별할 수 있습니다.
- 신원 확인 - 특별 입국 허가서를 소유하고 있는지 파악해야 하는 입국 키오스크 창구에서 유용합니다.

다음 단원: Azure의 Face 분석 시작하기

계속 >

< 이전

단위 2/5 ▼

다음 >

100 XP



Azure의 Face 분석 시작하기

3분

Microsoft Azure는 다음과 같이 얼굴을 감지 및 분석하는 데 사용할 수 있는 여러 인식 서비스를 제공합니다.

- **Computer Vision** 은 얼굴 감지와 연령 추측과 같은 기본적인 얼굴 분석을 제공합니다.
- **Video Indexer** 는 비디오에서 얼굴을 감지하고 식별하는 데 사용할 수 있습니다.
- **Face** 는 얼굴을 감지, 인식 및 분석할 수 있는 미리 빌드된 알고리즘을 제공합니다.

Face는 가장 폭넓은 얼굴 분석 기능을 제공하므로 이 모듈에서는 해당 서비스에 초점을 맞춥니다.

Face

Face는 현재 다음 기능을 제공합니다.

- 얼굴 감지
- 얼굴 확인
- 유사한 얼굴 찾기
- 유사성을 기준으로 얼굴 그룹화
- 사람 식별

Face는 이미지에서 발견되는 사람의 얼굴에 대한 사각형 좌표와 얼굴과 관련된 일련의 특성을 반환할 수 있습니다.

- **나이:** 나이 추측
- **흐림:** 얼굴의 흐릿한 정도(이미지의 초점이 얼굴에 있을 가능성을 나타낼 수 있음)
- **감정:** 표시되는 감정
- **노출:** 노출 부족 또는 노출 과다와 같은 요소이며 전체 이미지 노출이 아닌 이미지의 얼굴에 적용됨
- **수염:** 측정된 수염
- **안경:** 대상 인물이 안경을 쓰고 있는 경우
- **머리카락:** 머릿결 및 머리 색
- **머리 포즈:** 3D 공간에서의 얼굴 방향
- **화장:** 이미지에 보이는 얼굴에 화장을 했는지 여부
- **노이즈:** 이미지상 시각적 노이즈를 의미합니다. 어두운 설정을 위해 높은 ISO 설정으로 사진을 촬영한 경우 이미지에 노이즈가 보일 것입니다. 이미지가 거칠어 보이거나 명확성이 떨어지는 작은 점들로 이루어집니다.

- **폐색**: 이미지에서 얼굴을 가리는 요소가 있는지 확인
- **웃음**: 이미지 속 인물이 웃고 있는지 여부

Face용 Azure 리소스

Face를 사용하려면 Azure 구독에서 다음 유형의 리소스 중 하나를 만들어야 합니다.

- **얼굴**: 다른 Cognitive Services를 사용하지 않으려는 경우나 Face의 사용률과 비용을 별도로 추적하려는 경우 이 특정 리소스 유형을 사용하세요.
- **Cognitive Services**: 다른 많은 인지 서비스(Computer Vision, Text Analytics, Translator Text 등)와 함께 Computer Vision을 포함하는 일반적인 Cognitive Services 리소스입니다. 여러 인지 서비스를 사용할 계획이며 관리 및 개발을 단순화하려는 경우 이 리소스 유형을 사용하세요.

만들려는 리소스 유형과 관계없이 사용을 위해 필요한 두 가지 정보가 제공됩니다.

- 클라이언트 애플리케이션을 인증하는 데 사용되는 **키**입니다.
- 리소스에 액세스할 수 있는 HTTP 주소를 제공하는 **엔드포인트**입니다.

① 참고

Cognitive Services 리소스를 만드는 경우 클라이언트 응용 프로그램에서는 사용 중인 특정 서비스와 상관없이 동일한 키와 엔드포인트를 사용합니다.

보다 정확한 결과를 얻기 위한 팁

이미지에서 감지의 정확도를 향상하는 데 도움이 되는 몇 가지 고려 사항이 있습니다.

- 이미지 형식 - 지원되는 이미지는 JPEG, PNG, GIF 및 BMP입니다.
- 파일 크기 - 6MB 이하
- 얼굴 크기 범위 - 36x36~4096x4096. 더 작거나 더 큰 얼굴은 감지되지 않습니다.
- 기타 문제 - 얼굴 감지는 극단적인 얼굴 각도, 가림(선글라스 또는 손과 같은 얼굴을 가리는 물체)에 의해 제대로 작동하지 않을 수 있습니다. 얼굴이 정면 또는 정면에 최대한 가까울 때 최상의 결과를 얻을 수 있습니다.

다음 단원: 연습 - Face 서비스를 통해 얼굴 감지 및 분석

계속 >

< 이전

단위 3/5 ▼

다음 >

✓ 100 XP



연습 - Face 서비스를 통해 얼굴 감지 및 분석

15분

Face를 학습하는 가장 좋은 방법을 직접 살펴보는 것입니다.

시작하기 전에

이 연습을 완료하려면 다음이 필요합니다.

- Microsoft Azure 구독. 구독이 아직 없다면 <https://azure.microsoft.com/free>에서 평가판에 가입할 수 있습니다.
- MicrosoftDocs/ai-fundamentals GitHub 리포지토리에서 Jupyter Notebook을 실행할 수 있는 Python 환경. GitHub의 랩 파일을 자신의 Python 환경(있는 경우)에 복제하거나, 아래 지침에 따라 Azure 구독에서 Azure Machine Learning 작업 영역을 만들 수 있습니다.

① 참고

이 모듈은 Azure Machine Learning 작업 영역을 사용하는 여러 가지 모듈 중 하나입니다. **Azure AI 기본 사항** 인증의 준비 과정에서 이 모듈을 완료하는 경우, 작업 영역을 한 번 만들고 해당 인증 트랙의 모든 모듈에 다시 사용할 수 있습니다. 연습을 완료한 후에는 정리 지침에 따라 모듈 간 컴퓨팅 리소스를 중지해야 합니다.

Azure Machine Learning 작업 영역 만들기

Azure 구독에 Azure Machine Learning 작업 영역이 이미 있는 경우 새 브라우저 탭에서 [Azure Machine Learning Studio](#)로 이동하고 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다. 없는 경우 다음 단계에 따라 새 작업 영역을 만듭니다.

1. Azure 구독과 연결된 Microsoft 계정을 사용하여 [Azure Portal](#)에 로그인합니다.
2. + 리소스 만들기를 선택하고, *Machine Learning*을 검색한 뒤 다음 설정을 사용하여 새 *Machine Learning* 리소스를 만듭니다.
 - **작업 영역 이름:** 원하는 고유한 이름을 입력
 - **구독:** 자신의 Azure 구독
 - **리소스 그룹:** 고유한 이름의 새 리소스 그룹 만들기
 - **위치:** 사용 가능한 위치 선택

3. 작업 영역 리소스가 만들어질 때까지 기다립니다(몇 분이 걸릴 수 있음). 그런 다음 포털로 이동하고 작업 영역의 **개요** 페이지에서 Azure Machine Learning Studio를 실행(또는 새 브라우저 탭을 열고 <https://ml.azure.com> 으로 이동)한 다음 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다.
4. Azure Machine Learning Studio에서 왼쪽 상단에 있는 **☰** 아이콘을 설정/해제하면 인터페이스에 다양한 페이지가 표시됩니다. 해당 페이지를 사용하여 작업 영역에서 리소스를 관리할 수 있습니다.

컴퓨팅 인스턴스 만들기

이 연습에서 사용된 Notebook을 실행하려면 Azure Machine Learning 작업 영역에 컴퓨팅 인스턴스가 필요합니다. 컴퓨팅 인스턴스가 이미 있는 경우 시작하고, 없는 경우 다음 지침에 따라 새로 만듭니다.

1. [Azure Machine Learning Studio](#)에서 **관리** 아래에 있는 **컴퓨팅** 페이지를 확인합니다.
2. **컴퓨팅 인스턴스** 탭에서 다음 설정을 사용하여 새 컴퓨팅 인스턴스를 만듭니다.
 - **지역**: 사용 가능한 지역 선택
 - **가상 머신 유형**: CPU
 - **가상 머신 크기**: Standard_DS11_v2
 - **컴퓨팅 이름**: 고유한 이름 입력
3. 컴퓨팅 인스턴스가 시작될 때까지 기다립니다(1분 정도 걸릴 수 있음).

연습 파일 다운로드

이 모듈(및 기타 관련 모듈)에 사용되는 파일은 GitHub 리포지토리에 게시되는데, 이를 자신의 Python 환경에 복제해야 합니다. 이전 모듈에서 **ai-fundamentals** 리포지토리를 아직 복제하지 않은 경우 다음 단계를 사용하여 Azure Machine Learning 작업 영역에 복제합니다.

1. [Azure Machine Learning Studio](#)에서 **Notebooks** 페이지(**작성자** 아래)를 확인합니다. 이 페이지에는 Notebooks 실행에 사용할 수 있는 Notebook 편집기가 포함되어 있습니다.
2. **내 파일**에서 **□** 단추를 사용하여 다음 설정에 따라 새 파일을 만듭니다.
 - **파일 위치**: `Users/your user name`
 - **파일 이름**: Get-Files
 - **파일 형식**: Notebook
 - **이미 있는 경우 덮어쓰기**: 선택됨
3. 새 Notebook이 만들어지면 이전에 만든 컴퓨팅 인스턴스가 **컴퓨팅** 상자에서 선택되었는지, 그리고 상태가 **실행 중**인지 확인합니다. 그런 다음 Notebook에서 만든 사각형 셀에 다음 코드를 붙여넣습니다.

```
!git clone https://github.com/MicrosoftDocs/ai-fundamentals
```

4. 셀 옆에 있는 ▶ 단추를 사용하여 포함된 코드를 실행합니다. 이렇게 하면 GitHub에서 연습 파일이 복제됩니다.
5. 코드의 실행이 끝나고 파일의 체크아웃이 완료되면 내 파일 아래에 있는 ⌂ 단추를 사용하여 폴더 보기를 새로 고치고, **ai-fundamentals**라는 이름의 폴더가 만들어졌는지 확인합니다. 이 폴더에는 연습에 사용된 Notebook 및 기타 파일이 포함되어 있습니다.
6. Get-Files.ipynb Notebook 탭을 닫습니다.

연습 완료

Python 환경을 설정하고 **ai-fundamentals** 리포지토리를 복제했으면 얼굴 분석 및 인식을 살펴볼 준비가 된 것입니다.

1. **ai-fundamentals** 폴더에서 **Face Analysis.ipynb** Notebook을 엽니다. Azure Machine Learning Studio에서 Notebook 편집기를 사용하는 경우, <> 단추로 파일 탐색기 창을 축소하여 Notebook 탭에 집중할 수 있는 공간을 좀 더 마련합니다.
2. Notebook의 정보를 읽고 포함된 코드 셀을 순서대로 실행합니다.

정리

Azure Machine Learning Studio에서 컴퓨팅 인스턴스를 사용한 경우 Azure 크레딧을 불필요하게 사용하지 않도록 이를 중지해야 합니다.

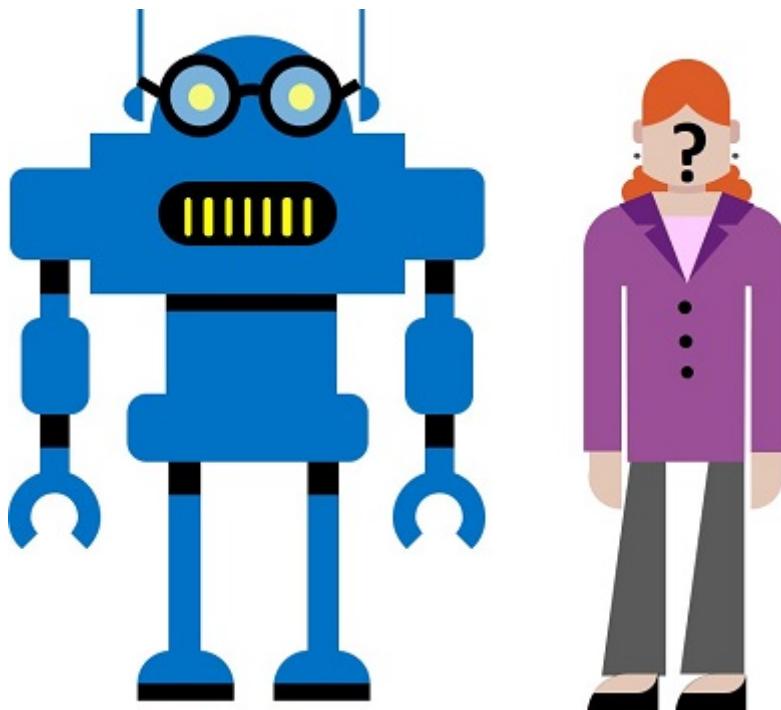
1. Azure Machine Learning Studio에서 관리 아래에 있는 컴퓨팅 페이지를 확인합니다.
2. 컴퓨팅 인스턴스 탭에서 컴퓨팅 인스턴스를 선택하고 중지 단추를 사용하여 중지합니다.

다음 단원: 지식 점검

계속 >

얼굴 감지 및 분석 Detecting and Analyzing Faces

Computer Vision 솔루션은 종종 사람의 얼굴을 감지, 분석 또는 식별할 수 있는 인공지능(AI) 솔루션이 필요하다. 예를 들어, 소매업체 NorthWind Traders가 AI 서비스를 통해 매장을 모니터링하여 도움이 필요한 고객을 파악하고 직원들에게 도움을 지시하는 '스마트 스토어'를 구현하기로 결정했다고 가정합시다. 이를 달성하기 위한 한 가지 방법은 얼굴 감지 및 분석을 수행하는 것이다. 즉, 이미지에 얼굴이 있는지 확인하고, 얼굴의 특징을 분석하는 것이다.



Face 서비스를 통해 얼굴 감지 및 분석

Northwind Traders 회사에서 원하는 스마트 시스템은 고객의 얼굴 특징을 검지하고 분석할 수 있는 도구라고 가정하자. Microsoft Azure에서는 Azure Cognitive 서비스의 일종인 **Face**를 활용하여 이것을 해결할 수 있다.

Cognitive 서비스 리소스 생성하기

Azure 구독에서 **Cognitive Services** 리소스를 생성하자.

알림: 만일 이미 Cognitive 서비스 리소스를 이미 가지고 있다면 Azure 포털에 있는 **최식 리소스**에 있는 리소스를 열어서 키와 엔드포인트를 복사해서 아래 셀에 붙여 넣으면 된다. 아니면 아래 순서로 만들면 된다

1. 브라우저의 새로운 탭을 열고, Azure portal(<https://portal.azure.com>)을 (<https://portal.azure.com>)을) 입력하고, Microsoft 계정으로 로그인한다..
2. + 리소스 만들기 버튼을 클릭하고, **Cognitive Services** 서비스를 찾은 다음, **Cognitive Services** 리소스를 다음과 같은 내용으로 생성한다.
 - **이름:** 유일한 이름을 입력한다(가능하면 영문과 숫자 사용).
 - **구독:** Azure 구독선택.
 - **위치:** 가능한 위치(한국 중부 추천):
 - **가격책정계층:** 표준 S0
 - **리소스 그룹:** 원하는 유리한 이름(가능하면 영문과 숫자 사용).

3. 배포가 완료될 때까지 기다린다. 그런 다음 Cognitive Services 리소스로 이동하여 *개요 페이지에서 링크를 클릭하여 서비스 키를 관리한다. 클라이언트 응용 프로그램에서 Cognitive Services 리소스에 연결하려면 엔드포인트와 키가 필요하다.

Cognitive Services 리소스에 있는 키와 엔드포인트 가져오기

Cognitive Services 리소스를 사용하기 위해서는, 클라이언트 응용 프로그램에서는 엔드포인트와 인증 키가 필요합니다. client applications need its endpoint and authentication key:

1. Azure portal에서, Cognitive Services 리소스를 선택하고 키 및 엔트포인트 페이지를 선택한 다음 키1을 복사하여 아래의 YOUR_COG_KEY를 붙여 넣는다.
2. 리소스에 있는 엔드포인트를 복사해서 아래의 YOUR_COG_ENDPOINT에 붙여 넣는다.

▲ 세이션에서 나오는 세이션의 이름과 함께 그대로 사용합니다

In []:

```
cog_key = 'YOUR_COG_KEY'
cog_endpoint = 'YOUR_COG_ENDPOINT'

print('Ready to use cognitive services at {} using key {}'.format(cog_endpoint, cog_key))
```

Cognitive Services 리소스에서 Face 서비스를 이용하기 위해서는 Azure Cognitive Services Face 패키지를 설치해야 한다.

In []:

```
! pip install azure-cognitiveservices-vision-face
```

이제 Cognitive 서비스 리소스와 설치된 SDK 패키지를 가지고 있으므로 가게 있는 사람 얼굴을 검지하기 위해 Face 서비스를 이용할 수 있다. 예를 확인하기 위해 아래 셀의 코드를 실행해보라.

In []:

```
from azure.cognitiveservices.vision.face import FaceClient
from msrest.authentication import CognitiveServicesCredentials
from python_code import faces
import os
%matplotlib inline

# 얼굴 검지 클라이언트를 만든다.
face_client = FaceClient(cog_endpoint, CognitiveServicesCredentials(cog_key))

# 이미지를 연다.
image_path = os.path.join('data', 'face', 'store_cam2.jpg')
image_stream = open(image_path, "rb")

# 얼굴을 검지한다.
detected_faces = face_client.face.detect_with_stream(image=image_stream)

# 얼굴을 나타낸다(python_code/faces.py의 코드를 이용한다)
faces.show_faces(image_path, detected_faces)
```

검지된 얼굴들은 각각 유일한 ID가 할당되고 응용 프로그램은 감지된 개별 얼굴을 구분할 수 있다.

아래 셀을 실행해서 쇼핑하고 있는 고객들의 얼굴에 대한 ID값을 확인한다.

In []:

```
# 이미지를 연다.
image_path = os.path.join('data', 'face', 'store_cam3.jpg')
image_stream = open(image_path, "rb")

# 얼굴을 감지한다.
detected_faces = face_client.face.detect_with_stream(image=image_stream)

# 얼굴들을 나타낸다.(yhton_code/faces.py에 있는 코드를 이용한다)
faces.show_faces(image_path, detected_faces, show_id=True)
```

얼굴의 속성들을 분석한다.

Face리소스는 단지 얼굴을 검지하는 것 외의 더 많은 일들을 수행할 수 있다. 얼굴 특징을 분석할 수 있고 나이와 감정상태에 등에 대한 설명을 제공한다. 예를들면 아래의 코드를 실행하여 쇼핑객의 얼굴 속성들을 분석해보자. For example, run the code below to analyze the facial attributes of a shopper.

In []:

```
# 이미지를 연다
image_path = os.path.join('data', 'face', 'store_cam1.jpg')
image_stream = open(image_path, "rb")

# 얼굴을 감지하고 얼굴 특성을 지정한다.
attributes = ['age', 'emotion']
detected_faces = face_client.face.detect_with_stream(image=image_stream, return_face_attributes=attributes)

# 얼굴들과 속성값들을 나타낸다(python_code/faces.py에 있는 코드를 이용한다).
faces.show_face_attributes(image_path, detected_faces)
```

이미지에있는 고객의 감정 상태 점수를 바탕으로 쇼핑 경험에 대한 행복정도를 판정할 수 있다.

비슷한 얼굴 찾기

검지된 각 얼굴에 대한 얼굴 ID 값들은 각 얼굴들끼리 일치도를 알아보는데 사용된다. 이런 ID를 이용하여 이전에 검지된 얼굴과 비교하는데 사용되고 비슷한 특징이 있는 얼굴들을 찾는데 사용된다.

예를들면 아래의 셀에 있는 코드를 실행하여 하나의 사진에 있는 쇼핑객과 다른 사진에 있는 모습을 비교하여 일치하는 얼굴을 찾는다.

In []:

```
# 이미지 1에 있는 첫번째 얼굴의 ID를 가져온다.
image_1_path = os.path.join('data', 'face', 'store_cam3.jpg')
image_1_stream = open(image_1_path, "rb")
image_1_faces = face_client.face.detect_with_stream(image=image_1_stream)
face_1 = image_1_faces[0]

# 두번째 이미지에 있는 Face ID 값들을 가져온다.
image_2_path = os.path.join('data', 'face', 'store_cam2.jpg')
image_2_stream = open(image_2_path, "rb")
image_2_faces = face_client.face.detect_with_stream(image=image_2_stream)
image_2_face_ids = list(map(lambda face: face.face_id, image_2_faces))

# 이미지 1에 있는 얼굴과 비슷한 얼굴이 이미지 2에 있는지 찾는다.
similar_faces = face_client.face.find_similar(face_id=face_1.face_id, face_ids=image_2_face_ids)

# 이미지 1에 있는 얼굴과 비슷한 얼굴을 이미지 2에서 찾아서 나타낸다(python_code/face.py에 있는 코드를 사용한다).
faces.show_similar_faces(image_1_path, face_1, image_2_path, image_2_faces, similar_faces)
```

얼굴 인식(Facial Recognition)하기

지금까지 여러분은 Face서비스로 얼굴과 얼굴의 특징을 감지할 수 있고, 서로 비슷한 두 얼굴을 식별할 수 있다는 것을 봤다. 특정 사람의 얼굴을 인식하도록 Face를 학습시키는 얼굴 인식(*Face Recognition*) 솔루션을 구현하면 한 단계 더 나아갈 수 있다. 이는 소셜 미디어 응용 프로그램에서 친구의 사진을 자동으로 태그하거나 생체 인식 확인 시스템의 일부로 얼굴 인식을 사용하는 등 다양한 시나리오에서 유용할 수 있다. 이런 작업들이 어떻게 일어나는지 살펴보기 위해 Northwind Traders 회사가 얼굴 인식서비스를 이용하여 IT 부서의 권한이 있는 직원만 보안 시스템에 액세스하도록 하는 시스템을 만들고 싶다고 가정한다.

권한이 있는 직원을 대표할 수 있는 직원그룹을 만드는 작업부터 시작해보자.

In []:

```
group_id = 'employee_group_id'
try:
    # Delete group if it already exists
    face_client.person_group.delete(group_id)
except Exception as ex:
    print(ex.message)
finally:
    face_client.person_group.create(group_id, 'employees')
    print ('Group created!')
```

직원 그룹이 존재하므로 해당 그룹에 포함시키기 원하는 직원들을 그룹에 포함시킨다. 그리고 나서 각 직원들에 대한 여러종류의 사진들을 등록하여 Face서비스가 각 사람들의 독특한 특징들에 대한 것들을 학습하도록 한다. 이상적인 것은 특정사람이 다양한 포즈와 서로다른 감정을 나타내는 사진이 필요하다. 우리는 Wendell이라는 한명의 직원을 등록하고 해당 직원에 대한 여러 종류의 사진을 등록할 것이다.

In []:

```

import matplotlib.pyplot as plt
from PIL import Image
import os
%matplotlib inline

# 직원(Wendell)을 그룹에 추가한다.
wendell = face_client.person_group_person.create(group_id, 'Wendell')

# Wendell에 대한 사진들을 가져온다
folder = os.path.join('data', 'face', 'wendell')
wendell_pics = os.listdir(folder)

# 사진들을 등록한다.
i = 0
fig = plt.figure(figsize=(8, 8))
for pic in wendell_pics:
    # 직원 그룹이 있는 사람에게 각 사진들을 추가한다.
    img_path = os.path.join(folder, pic)
    img_stream = open(img_path, "rb")
    face_client.person_group_person.add_face_from_stream(group_id, wendell.person_id, img_stream)
)

# 각 이미지들을 표시한다.
img = Image.open(img_path)
i += 1
a=fig.add_subplot(1, len(wendell_pics), i)
a.axis('off')
imgplot = plt.imshow(img)
plt.show()

```

추가된 직원과 사진들을 등록하였다면 Face를 학습하여 각 사진을 인식하도록 할 수 있다.

In []:

```

face_client.person_group.train(group_id)
print('Trained!')

```

이제 학습된 모델을 사용하여 이미지에 있는 얼굴들을 구분하는데 사용할 수 있다.

In []:

```
# 두번째 이미지에 있는 얼굴 ID를 가져온다.
image_path = os.path.join('data', 'face', 'employees.jpg')
image_stream = open(image_path, "rb")
image_faces = face_client.face.detect_with_stream(image=image_stream)
image_face_ids = list(map(lambda face: face.face_id, image_faces))

# 인식한 얼굴이름들을 가져온다
face_names = {}
recognized_faces = face_client.face.identify(image_face_ids, group_id)
for face in recognized_faces:
    person_name = face_client.person_group_person.get(group_id, face.candidates[0].person_id).name
    face_names[face.face_id] = person_name

# 인식한 얼굴들을 나타낸다.
faces.show_recognized_faces(image_path, image_faces, face_names)
```

심화 학습

Face 인지 서비스에 대하여 더 많이 알고 싶다면 [Face documentation](https://docs.microsoft.com/azure/cognitive-services/face/) (<https://docs.microsoft.com/azure/cognitive-services/face/>) 를 참조하라.

In []:



✓ 600 XP ➔

Computer Vision 서비스로 텍스트 읽기

24분 • 모듈 • 5 단위

★★★★★ 4.8 (608)

평가하기

초급 AI 엔지니어 데이터 과학자 개발자 솔루션 아키텍처 학생

Cognitive Services

OCR(광학 인식)을 통해 AI(인공 지능) 시스템에서는 이미지의 텍스트를 읽을 수 있으므로 애플리케이션에서 사진, 스캔한 문서 및 기타 디지털 텍스트 원본에서 정보를 추출할 수 있습니다.

학습 목표

Computer Vision 서비스를 사용하여 이미지의 텍스트를 읽는 방법 알아보기

책갈피 + 컬렉션에 추가

필수 조건

Azure Portal 탐색 능력

이 모듈은 이러한 학습 경로의 일부입니다.

[Microsoft Azure에서 컴퓨터 비전 살펴보기](#)

소개

3분



Azure에서 OCR 시작

3분



연습 - Computer Vision 서비스로 텍스트 읽기

15분



지식 점검

2분



요약

1분



✓ 100 XP



소개

3분

컴퓨터 시스템이 필기하거나 인쇄 텍스트를 처리하는 기능은 *Computer Vision* 이 자연어 처리와 교차하는 AI(인공 지능)의 영역입니다. 텍스트를 "읽기" 위해 Computer Vision 기능이 필요하며 이를 이해하기 위해 자연어 처리 기능이 필요합니다.

인쇄된 텍스트를 처리하는 기본 토대는 개별 모양을 문자, 숫자, 문장 부호 또는 텍스트의 다른 요소로 인식하도록 모델을 학습할 수 있는 OCR(광학 문자 인식)입니다. 이러한 종류의 기능을 구현하는 초기 작업의 상당수는 우편 번호에 따라 메일의 자동 정렬을 지원하는 우편 서비스로 실시되었습니다. 그 이후로 텍스트를 읽기 위한 최신 기술이 계속 진행되었으며, 이제 이미지에서 인쇄 또는 필기 텍스트를 감지하고 줄별 또는 단어별로 읽을 수 있는 모델을 빌드할 수 있습니다.

다른 측면에는 AI 시스템이 텍스트 문자를 읽을 수 있을 뿐만 아니라 의미 체계 모델을 사용하여 텍스트를 해석하는 데 사용되는 MRC(기계 읽기 이해)가 있습니다.

이 모듈에서는 OCR 기술을 사용하여 이미지에서 텍스트를 검색하고 텍스트 기반 데이터 형식으로 변환한 다음, 추가 처리 또는 분석을 위해 저장, 인쇄 또는 입력으로 사용할 수 있는 방법을 집중적으로 설명합니다.

OCR 사용 사례

이미지에서 인쇄 및 필기 텍스트를 인식하는 기능은 다음과 같은 많은 시나리오에서 유용합니다.

- 필기
- 의료 기록이나 기록 문서와 같은 양식 디지털화
- 은행 예금을 위해 인쇄하거나 손으로 쓴 수표 스캔

다음 단원: Azure에서 OCR 시작

[계속 >](#)

< 이전

단위 2/5 ▼

다음 >

✓ 100 XP



Azure에서 OCR 시작

3분

이미지에서 텍스트를 추출하는 기능은 이미지 분석 기능도 제공하는 Computer Vision 서비스에서 처리됩니다.

Computer Vision용 Azure 리소스

Computer Vision 서비스를 사용하는 첫 번째 단계는 Azure 구독에서 리소스를 만드는 것입니다. 다음 리소스 유형 중 하나를 사용할 수 있습니다.

- **Computer Vision** : Computer Vision 서비스에 대한 특정 리소스입니다. 다른 인지 서비스를 사용하지 않으려는 경우 또는 Computer Vision 리소스의 사용률과 비용을 별도로 추적 하려는 경우 이 리소스 유형을 사용하세요.
- **Cognitive Services** : Text Analytics 및 Translator Text 등의 다른 많은 인지 서비스와 함께 Computer Vision을 포함하는 일반적인 인지 서비스 리소스입니다. 여러 인지 서비스를 사용할 계획이며 관리 및 개발을 단순화하려는 경우 이 리소스 유형을 사용하세요.

만들려는 리소스 유형과 관계없이 사용을 위해 필요한 두 가지 정보가 제공됩니다.

- 클라이언트 애플리케이션을 인증하는 데 사용되는 **키**입니다.
- 리소스에 액세스할 수 있는 HTTP 주소를 제공하는 **엔드포인트**입니다.

① 참고

Cognitive Services 리소스를 만드는 경우 클라이언트 애플리케이션에서는 사용 중인 특정 서비스와 상관없이 동일한 키와 엔드포인트를 사용합니다.

Computer Vision 서비스를 사용하여 텍스트 읽기

이미지에 텍스트가 여러 번 포함됩니다. 타이핑한 텍스트 또는 손으로 쓴 텍스트일 수 있습니다. 일반적인 예로는 도로 표지판이 있는 이미지, JPEG나 PNG 파일 형식과 같은 이미지 형식의 스캔 문서 또는 회의 중에 사용된 화이트보드에서 찍은 사진일 수도 있습니다.

Computer Vision 서비스는 **OCR API**와 **읽기 API**라는 이미지에서 텍스트를 읽는 데 사용할 수 있는 두 가지 API(애플리케이션 프로그래밍 인터페이스)를 제공합니다.

OCR API

OCR API는 이미지에서 소량의 텍스트를 빠르게 추출하도록 설계되었습니다. 즉각적인 결과를 제공하기 위해 동기적으로 작동하며 여러 언어로 텍스트를 인식할 수 있습니다.

OCR API를 사용하여 이미지를 처리하면 다음으로 구성된 정보 계층 구조가 반환됩니다.

- 텍스트가 포함된 이미지의 **영역**
- 각 영역의 텍스트 **줄**
- 각 텍스트 줄의 **단어**

이러한 각 요소에 대해 OCR API는 지역, 줄 또는 단어가 표시되는 이미지의 위치를 나타내기 위해 직사각형을 정의하는 경계 상자 좌표를 반환합니다.

읽기 API

OCR 메서드는 이미지가 텍스트가 많은 것으로 간주될 때 가양성의 문제가 있을 수 있습니다. 읽기 API는 최신 인식 모델을 사용하며 상당한 양의 텍스트가 있거나 상당한 시각적 노이즈가 있는 이미지에 최적화되어 있습니다.

텍스트가 많은 스캔한 문서에는 읽기 API가 더 좋습니다. 읽기 API는 텍스트 줄을 고려하고 인쇄된 텍스트로 이미지를 지원하고 필기를 인식하여 사용할 적절한 인식 모델을 자동으로 결정할 수 있습니다.

읽기 API는 더 큰 문서에서 작동할 수 있으므로 콘텐츠를 읽고 결과를 애플리케이션에 반환하는 동안 애플리케이션을 차단하지 않도록 비동기적으로 작동합니다. 즉, 읽기 API를 사용하려면 애플리케이션에서 다음 3단계 프로세스를 사용해야 합니다.

1. API에 이미지를 제출하고 응답으로 작업 *ID*를 검색합니다.
2. 작업 *ID*를 사용하여 이미지 분석 작업의 상태를 확인하고 완료될 때까지 기다립니다.
3. 작업의 결과를 검색합니다.

읽기 API의 결과는 다음 계층 구조로 정렬됩니다.

- **페이지** - 페이지 크기 및 방향에 관한 정보를 포함한 텍스트의 각 페이지입니다.
- **줄** - 한 페이지의 텍스트 줄 수입니다.
- **단어** - 텍스트 한 줄의 단어 수입니다.

각 줄과 단어에는 페이지 상의 위치를 나타내는 경계 상자 좌표가 포함됩니다.

다음 단원: 연습 - Computer Vision 서비스로 텍스트 읽기

계속 >

< 이전

단위 3/5 ▼

다음 >

100 XP



연습 - Computer Vision 서비스로 텍스트 읽기

15분

Computer Vision 서비스를 사용하여 텍스트를 읽는 방법을 배우려면 직접 살펴보는 것이 가장 좋습니다.

시작하기 전에

이 연습을 완료하려면 다음이 필요합니다.

- Microsoft Azure 구독. 구독이 아직 없다면 <https://azure.microsoft.com/free>에서 평가판에 가입할 수 있습니다.
- MicrosoftDocs/ai-fundamentals GitHub 리포지토리에서 Jupyter Notebook을 실행할 수 있는 Python 환경. GitHub의 랩 파일을 자신의 Python 환경(있는 경우)에 복제하거나, 아래 지침에 따라 Azure 구독에서 Azure Machine Learning 작업 영역을 만들 수 있습니다.

! 참고

이 모듈은 Azure Machine Learning 작업 영역을 사용하는 여러 가지 모듈 중 하나입니다.

Azure AI 기본 사항 인증의 준비 과정에서 이 모듈을 완료하는 경우, 작업 영역을 한 번 만들고 해당 인증 트랙의 모든 모듈에 다시 사용할 수 있습니다. 연습을 완료한 후에는 정리 지침에 따라 모듈 간 컴퓨팅 리소스를 중지해야 합니다.

Azure Machine Learning 작업 영역 만들기

Azure 구독에 Azure Machine Learning 작업 영역이 이미 있는 경우 새 브라우저 탭에서 [Azure Machine Learning Studio](#)로 이동하고 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다. 없는 경우 다음 단계에 따라 새 작업 영역을 만듭니다.

1. Azure 구독과 연결된 Microsoft 계정을 사용하여 [Azure Portal](#)에 로그인합니다.
2. + 리소스 만들기를 선택하고, *Machine Learning*을 검색한 뒤 다음 설정을 사용하여 새 *Machine Learning* 리소스를 만듭니다.
 - **작업 영역 이름:** 원하는 고유한 이름을 입력
 - **구독:** 자신의 Azure 구독
 - **리소스 그룹:** 고유한 이름의 새 리소스 그룹 만들기
 - **위치:** 사용 가능한 위치 선택

3. 작업 영역 리소스가 만들어질 때까지 기다립니다(몇 분이 걸릴 수 있음). 그런 다음 포털로 이동하고 작업 영역의 **개요** 페이지에서 Azure Machine Learning Studio를 실행(또는 새 브라우저 탭을 열고 <https://ml.azure.com> 으로 이동)한 다음 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다.
4. Azure Machine Learning Studio에서 왼쪽 상단에 있는 **☰** 아이콘을 설정/해제하면 인터페이스에 다양한 페이지가 표시됩니다. 해당 페이지를 사용하여 작업 영역에서 리소스를 관리할 수 있습니다.

컴퓨팅 인스턴스 만들기

이 연습에서 사용된 Notebook을 실행하려면 Azure Machine Learning 작업 영역에 컴퓨팅 인스턴스가 필요합니다. 컴퓨팅 인스턴스가 이미 있는 경우 시작하고, 없는 경우 다음 지침에 따라 새로 만듭니다.

1. [Azure Machine Learning Studio](#)에서 **관리** 아래에 있는 **컴퓨팅** 페이지를 확인합니다.
2. **컴퓨팅 인스턴스** 탭에서 다음 설정을 사용하여 새 컴퓨팅 인스턴스를 만듭니다.
 - **지역**: 사용 가능한 지역 선택
 - **가상 머신 유형**: CPU
 - **가상 머신 크기**: Standard_DS11_v2
 - **컴퓨팅 이름**: 고유한 이름 입력
3. 컴퓨팅 인스턴스가 시작될 때까지 기다립니다(1분 정도 걸릴 수 있음).

연습 파일 다운로드

이 모듈(및 기타 관련 모듈)에 사용되는 파일은 GitHub 리포지토리에 게시되는데, 이를 자신의 Python 환경에 복제해야 합니다. 이전 모듈에서 **ai-fundamentals** 리포지토리를 아직 복제하지 않은 경우 다음 단계를 사용하여 Azure Machine Learning 작업 영역에 복제합니다.

1. [Azure Machine Learning Studio](#)에서 **Notebooks** 페이지(**작성자** 아래)를 확인합니다. 이 페이지에는 Notebooks 실행에 사용할 수 있는 Notebook 편집기가 포함되어 있습니다.
2. **내 파일**에서 **□** 단추를 사용하여 다음 설정에 따라 새 파일을 만듭니다.
 - **파일 위치**: `Users/your user name`
 - **파일 이름**: Get-Files
 - **파일 형식**: Notebook
 - **이미 있는 경우 덮어쓰기**: 선택됨
3. 새 Notebook이 만들어지면 이전에 만든 컴퓨팅 인스턴스가 **컴퓨팅** 상자에서 선택되었는지, 그리고 상태가 **실행 중**인지 확인합니다. 그런 다음 Notebook에서 만든 사각형 셀에 다음 코드를 붙여넣습니다.

```
!git clone https://github.com/MicrosoftDocs/ai-fundamentals
```

4. 셀 옆에 있는 ▶ 단추를 사용하여 포함된 코드를 실행합니다. 이렇게 하면 GitHub에서 연습 파일이 복제됩니다.
5. 코드의 실행이 끝나고 파일의 체크아웃이 완료되면 내 파일 아래에 있는 ⌂ 단추를 사용하여 폴더 보기를 새로 고치고, **ai-fundamentals**라는 이름의 폴더가 만들어졌는지 확인합니다. 이 폴더에는 연습에 사용된 Notebook 및 기타 파일이 포함되어 있습니다.
6. Get-Files.ipynb Notebook 탭을 닫습니다.

연습 완료

Python 환경을 설정하고 **ai-fundamentals** 리포지토리를 복제했으면 광학 인식을 살펴볼 준비가 된 것입니다.

1. **ai-fundamentals** 폴더에서 **Optical Character Recognition.ipynb** Notebook을 엽니다. Azure Machine Learning Studio에서 Notebook 편집기를 사용하는 경우, <> 단추로 파일 탐색기 창을 축소하여 Notebook 탭에 집중할 수 있는 공간을 좀 더 마련합니다.
2. Notebook의 정보를 읽고 포함된 코드 셀을 순서대로 실행합니다.

정리

Azure Machine Learning Studio에서 컴퓨팅 인스턴스를 사용한 경우 Azure 크레딧을 불필요하게 사용하지 않도록 이를 중지해야 합니다.

1. Azure Machine Learning Studio에서 관리 아래에 있는 컴퓨팅 페이지를 확인합니다.
2. 컴퓨팅 인스턴스 탭에서 컴퓨팅 인스턴스를 선택하고 중지 단추를 사용하여 중지합니다.

다음 단원: 지식 점검

계속 >

광학 문자 인식(Optical Character Recognition)



일반적인 컴퓨터 비전 과제는 이미지에서 텍스트를 감지하고 해석하는 것이다. 이러한 종류의 처리를 흔히 광학 문자 인식(OCR)이라고 한다.

Computer Vision 서비스를 이용하여 이미지에 있는 텍스트 읽기

Computer Vision Cognitive 서비스는 다음과 같은 OCR 작업을 위한 기능을 제공해준다.

- **OCR API**를 이용하여 여러 언어로 된 텍스트를 읽는다. 이 API는 동기식으로 사용할 수 있으며, 이미지에서 소량의 텍스트를 감지하고 읽어야 할 때 잘 작동한다.
- **Read API**는 대용량 문서에 최적화되어 있다. 이 API는 비동기식으로 사용되며 인쇄 텍스트와 필기 텍스트 모두에 사용할 수 있다

이 서비스는 **Computer Vision**나 **Cognitive Services** 리소스를 생성하여 사용할 수 있다.

만일 아직 이 서비스가 없다면 Azure 구독에서 **Cognitive Services** 리소스를 생성할 수 있다.

1. 브라우저의 새로운 탭을 열고, Azure portal(<https://portal.azure.com>)을 (<https://portal.azure.com>)을) 입력하고, Microsoft 계정으로 로그인한다..
2. + 리소스 만들기 버튼을 클릭하고, **Cognitive Services** 서비스를 찾은 다음, **Cognitive Services** 리소스를 다음과 같은 내용으로 생성한다.
 - **이름**: 유일한 이름을 입력한다(가능하면 영문과 숫자 사용).
 - **구독**: Azure 구독선택.
 - **위치**: 가능한 위치(한국 중부 추천):
 - **가격 책정 계층**: 표준 S0
 - **리소스 그룹**: 원하는 유리한 이름(가능하면 영문과 숫자 사용).
3. 배포가 완료될 때까지 기다린다. 그런 다음 Cognitive Services 리소스로 이동하여 *개요 페이지에서 링크를 클릭하여 서비스 키를 관리한다. 클라이언트 응용 프로그램에서 Cognitive Services 리소스에 연결하려면 앤드포인트과 키가 필요하다.

Cognitive Services 리소스에 있는 키와 엔드포인트 가져오기

Cognitive Services 리소스를 사용하기 위해서는, 클라이언트 응용프로그램에서는 엔드포인트와 인증 키가 필요하다.

1. Azure portal에서, Cognitive Services 리소스를 선택하고 **키 및 엔트포인트** 페이지를 선택한 다음 **키1** 을 복사하여 아래의 **YOUR_COG_KEY**.를 붙여 넣는다.
2. 리소스에 있는 **엔드포인트** 를 복사해서 아래의 **YOUR_COG_ENDPOINT**.에 붙여 넣는다.

▲ 세이션 탭에서 세션 ID로 시작되는 URL은 컴퓨터에 의해 그대로 사용된다.

In []:

```
cog_key = 'YOUR_COG_KEY'  
cog_endpoint = 'YOUR_COG_ENDPOINT'  
  
print('Ready to use cognitive services at {} using key {}'.format(cog_endpoint, cog_key))
```

이제 키와 엔드포인트를 설정했으니 이미지로부터 텍스트를 추출하기 위해 Computer Vision 서비스 리소스를 사용할 수 있다.

Python에서 이런 작업을 하기 위해서는 아래의 셀을 실행하여 the Azure Cognitive Services Computer Vision 패키지를 실행한다.

In []:

```
! pip install azure-cognitiveservices-vision-computervision
```

이제 Computer Vision 서비스를 이용하여 이미지에 있는 텍스트를 읽을 수 있을 준비가 되어 있다.

OCR API로 시작해보자. 이 API를 사용하면 이미지를 동기적으로 분석하고 포함된 텍스트를 읽을 수 있다. 이 경우, 가상의 Northwind Traders 소매 회사의 광고 이미지에 포함된 텍스트를 확인할 수 있다.

In []:

```

from azure.cognitiveservices.vision.computervision import ComputerVisionClient
from msrest.authentication import CognitiveServicesCredentials
import matplotlib.pyplot as plt
from PIL import Image, ImageDraw
import os
%matplotlib inline

# Computer Vision 서비스에 접근하기 위한 클라이언트를 가져온다.
computervision_client = ComputerVisionClient(cog_endpoint, CognitiveServicesCredentials(cog_key))

# 이미지 파일을 읽는다.
image_path = os.path.join('data', 'ocr', 'advert.jpg')
image_stream = open(image_path, "rb")

# Computer Vision 서비스를 이용하여 이미지에 포함된 텍스트를 가져온다.
read_results = computervision_client.read_printed_text_in_stream(image_stream)

# 텍스트를 한줄씩 처리한다
for region in read_results.regions:
    for line in region.lines:

        # 텍스트 줄에 있는 단어별로 가져온다.
        line_text = ''
        for word in line.words:
            line_text += word.text + ' '
        print(line_text.rstrip())

# 이미지를 화면에 나타낸다.
fig = plt.figure(figsize=(7, 7))
img = Image.open(image_path)
draw = ImageDraw.Draw(img)
plt.axis('off')
plt.imshow(img)

```

이미지에서 발견된 텍스트는 영역, 선 및 단어의 계층 구조로 구성되고 코드는 결과를 검색하기 위해 이를 읽는다.

그 결과 이미지 위에 있는 텍스트를 볼 수 있다.

경계 상자(Bounding Box) 나타내기

또한 결과에는 이미지에 있는 텍스트 행과 개별 단어에 대한 경계 상자의 좌표가 포함됩니다. 위에서 검색한 광고 이미지의 텍스트 행에 대한 경계 상자를 보려면 아래 셀을 실행하라.

In []:

```
# 이미지를 표시하기위에 연다.
fig = plt.figure(figsize=(7, 7))
img = Image.open(image_path)
draw = ImageDraw.Draw(img)

# 텍스트를 줄별로 처리한다.
for region in read_results.regions:
    for line in region.lines:

        # 각 줄에서 텍스트의 위치를 나타낸다.
        l,t,w,h = list(map(int, line.bounding_box.split(',')))
        draw.rectangle(((l,t), (l+w, t+h)), outline='magenta', width=5)

        # 텍스트 줄에 있는 단어를 읽는다.
        line_text = ''
        for word in line.words:
            line_text += word.text + ' '
        print(line_text.rstrip())

# 텍스트의 위치를 강조한 이미지를 보여준다.
plt.axis('off')
plt.imshow(img)
```

그 결과 텍스트의 각 라인에 대한 경계상자가 이미지 위에 사각형으로 나타난다.

Read API 사용하기

이전에 사용한 OCR API는 적은 양의 텍스트가 있는 이미지에 적합하다. 스캔한 문서와 같이 더 많은 텍스트 본문을 읽어야 할 경우 **Read API**를 사용할 수 있다. 그 과정은 다음과 같이 여러 단계가 필요하다.:

1. 이미지를 Computer Vision 서비스로 보내어 읽고 비동기적으로 분석하도록 한다.
2. 분석작업이 완료될때까지 기다린다.
3. 분석결과 값을 받아서 확인한다.

아래의 셀을 실행함으로 Northwind Traders 가게의 메니저에게 보낼 스캔한 편지 속에 있는 텍스트를 읽는 작업을 수행한다.

In []:

```

from azure.cognitiveservices.vision.computervision import ComputerVisionClient
from azure.cognitiveservices.vision.computervision.models import OperationStatusCodes
from msrest.authentication import CognitiveServicesCredentials
import matplotlib.pyplot as plt
from PIL import Image
import time
import os
%matplotlib inline

# 이미지 파일을 읽는다.
image_path = os.path.join('data', 'ocr', 'letter.jpg')
image_stream = open(image_path, "rb")

# Computer Vision 서비스를 사용하기 위한 클라이언트를 가져온다.
computervision_client = ComputerVisionClient(cog_endpoint, CognitiveServicesCredentials(cog_key))

# 이미지에 프린트되어 있는 텍스트를 읽고 작업 ID를 가져오기 위한 요청을 보낸다.
read_operation = computervision_client.read_in_stream(image_stream,
                                                       raw=True)
operation_location = read_operation.headers["Operation-Location"]
operation_id = operation_location.split("/")[-1]

# 비동기적인 작업이 완료될때까지 기다린다.
while True:
    read_results = computervision_client.get_read_result(operation_id)
    if read_results.status not in [OperationStatusCodes.running]:
        break
    time.sleep(1)

# 만일 작업이 완성되었다면 텍스트를 줄단위로 처리한다.
if read_results.status == OperationStatusCodes.succeeded:
    for result in read_results.analyze_result.read_results:
        for line in result.lines:
            print(line.text)

# 이미지를 열고 나타낸다.
print('Wn')
fig = plt.figure(figsize=(12,12))
img = Image.open(image_path)
plt.axis('off')
plt.imshow(img)

```

결과를 살펴보라. 편지의 전체 복사본이 있는데 이것은 손으로 쓴 사인과 함께 대부분은 인쇄된 텍스트로 구성되어 있다. 편지의 원래 임자는 OCR결과 밑에 나타나게 된다(내용을 보기 위해서 스크롤 할 필요가 있다)

손으로 쓴 텍스트 읽기

이전 예제에서 이미지 분석 요청은 프린트된 텍스트에 대한 텍스트 인식 작업에 최적화 되어 있었다. 그럼에도 불구하고 손으로 쓴 서명이 읽혀졌다는 점에 유의하자.

손으로 쓴 글을 읽는 능력은 매우 유용하다. 예를 들어, 수기로 쓴 쇼핑 목록이 포함된 노트를 작성한 경우 스마트폰 앱을 사용하여 노트를 읽어 노트에 포함된 텍스트를 찾아내기를 원한다고 가정하자.

아래의 셀을 실행하여 수기로 작성한 쇼핑 목록에 대한 읽기 작업의 예를 확인하라.

In []:

```

from azure.cognitiveservices.vision.computervision import ComputerVisionClient
from azure.cognitiveservices.vision.computervision.models import OperationStatusCodes
from msrest.authentication import CognitiveServicesCredentials
import matplotlib.pyplot as plt
from PIL import Image
import time
import os
%matplotlib inline

# 이미지 파일을 읽는다.
image_path = os.path.join('data', 'ocr', 'note.jpg')
image_stream = open(image_path, "rb")

# 컴퓨터 비전 서비스를 위한 클라이언트를 가져온다.
computervision_client = ComputerVisionClient(cog_endpoint, CognitiveServicesCredentials(cog_key))

# 이미지에 포함되어 있는 프린트된 텍스트를 읽기 위한 요청을 보내고 operation ID를 가져온다
read_operation = computervision_client.read_in_stream(image_stream,
                                                       raw=True)
operation_location = read_operation.headers["Operation-Location"]
operation_id = operation_location.split("/")[-1]

# 비동기적인 작업이 완료될 때까지 기다린다.
while True:
    read_results = computervision_client.get_read_result(operation_id)
    if read_results.status not in [OperationStatusCodes.running]:
        break
    time.sleep(1)

# 만일 작업이 성공적으로 끝났다면 줄 단위로 텍스트를 읽는다.
if read_results.status == OperationStatusCodes.succeeded:
    for result in read_results.analyze_result.read_results:
        for line in result.lines:
            print(line.text)

# 이미지를 열어서 나타낸다.
print('Wn')
fig = plt.figure(figsize=(12,12))
img = Image.open(image_path)
plt.axis('off')
plt.imshow(img)

```

더 많은 정보

OCR을 위한 Computer Vision 서비스의 사용에 대한 더 많은 정보를 여기를 확인하라 [Computer Vision 문서](https://docs.microsoft.com/ko-kr/azure/cognitive-services/computer-vision/concept-recognizing-text) (<https://docs.microsoft.com/ko-kr/azure/cognitive-services/computer-vision/concept-recognizing-text>)



✓ 600 XP ➔

Form Recognizer 서비스를 사용하여 영수증 분석

24분 • 모듈 • 5 단위

★★★★★ 4.8 (793)

평가하기

초급 AI 엔지니어 데이터 과학자 개발자 솔루션 아키텍처 학생

Cognitive Services

청구서 및 영수증 처리는 많은 비즈니스 시나리오에서 일반적인 작업입니다. 점점 더 많은 조직이 AI(인공 지능)를 활용하여 스캔한 영수증으로부터의 데이터 추출을 자동화하고 있습니다.

학습 목표

Form Recognizer 서비스에서 기본 제공되는 영수증 처리 기능을 사용하는 방법을 알아봅니다.

📘 책갈피 Ⓛ 컬렉션에 추가

필수 조건

Azure Portal 탐색 능력

이 모듈은 이러한 학습 경로의 일부입니다.

[Microsoft Azure에서 컴퓨터 비전 살펴보기](#)

소개

3분



Azure에서 영수증 분석 시작

3분



연습 - Form Recognizer를 사용하여 영수증 분석

15분



지식 점검

2분



요약

1분



✓ 100 XP



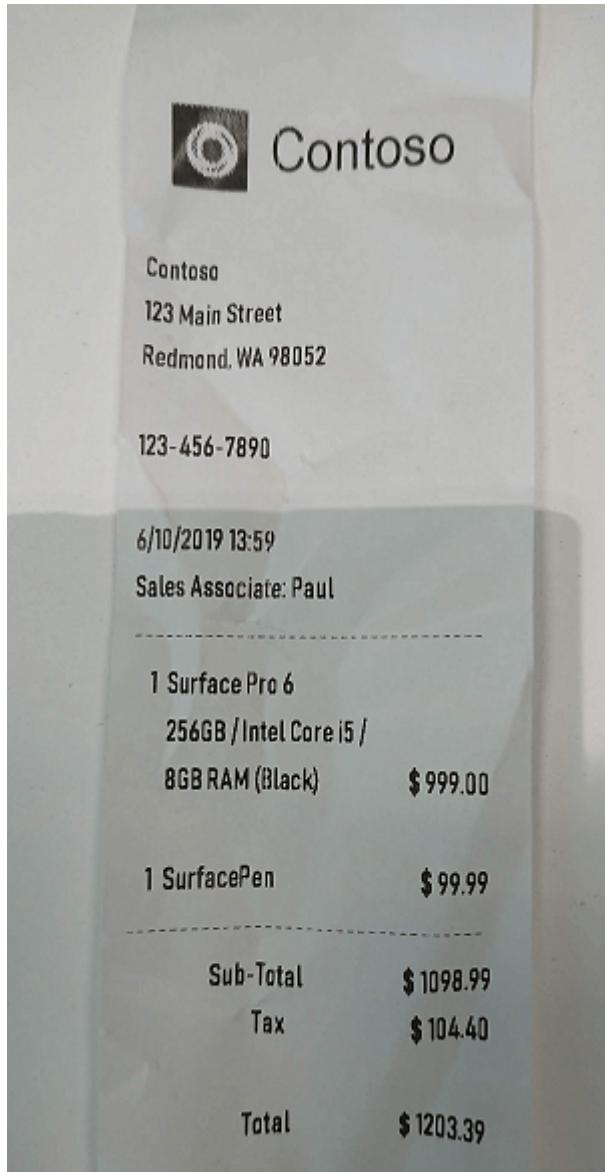
소개

3분

많은 조직에서 흔히 발생하는 문제는 수신 또는 청구서 데이터를 처리해야 한다는 것입니다. 예를 들어, 회사는 스캔한 영수증을 사용하여 비용 청구를 전자적으로 제출해야 하거나 송장을 디지털화하고 올바른 계정 부서로 라우팅해야 할 수 있습니다.

영수증을 스캔하여 디지털 이미지나 PDF 문서를 만드는 것은 비교적 간단하며, OCR(광학 문자 인식) 기술을 사용하여 디지털화된 문서에서 텍스트 내용을 추출할 수 있습니다. 그러나 일반적으로 포함된 정보를 이해하려면 추출된 텍스트를 검토해야 합니다.

다음 영수증을 살펴보세요.



영수증에는 비용 청구에 필요할 수 있는 다음과 같은 정보가 포함되어 있습니다.

- 판매자의 이름, 주소 및 전화번호.
- 구매 날짜 및 시간.
- 구매한 각 항목의 수량과 가격.
- 하위 합계, 세금 및 총 금액.

점점 더 많은 조직이 많은 양의 영수증과 송장을 처리할 때 영수증에서 텍스트 데이터를 추출할 뿐만 아니라 포함된 정보를 지능적으로 해석할 수 있는 AI(인공 지능) 솔루션을 찾고 있습니다.

다음 단원: Azure에서 영수증 분석 시작

계속 >

< 이전

단위 2/5 ▼

다음 >

✓ 100 XP



Azure에서 영수증 분석 시작

3분

Azure의 **Form Recognizer**는 양식, 송장 및 영수증과 같은 문서에서 데이터 처리를 자동화하는데 사용할 수 있는 지능형 양식 처리 기능을 제공합니다. 다음과 같은 방법을 통해 양식 데이터를 해석할 수 있는 예측 모델과 최첨단 OCR (광학 문자 인식)을 결합합니다.

- 필드 이름을 값에 일치.
- 데이터 테이블 처리.
- 날짜, 전화번호, 주소, 합계 등과 같은 특정 유형의 필드 식별.

Form Recognizer는 다음과 같은 자동화된 문서 처리를 지원합니다.

- **사전 빌드된 영수증 모델**은 즉시 제공되며 판매 영수증에서 데이터를 인식하고 추출하도록 학습된 것입니다.
- 양식에서 키/값 쌍 및 테이블 데이터로 알려진 내용을 추출할 수 있는 **사용자 지정 모델**. 사용자 지정 모델은 사용자 고유의 데이터를 사용하도록 학습되어 특정 양식에 맞게 이 모델을 조정하는 데 도움이 됩니다. 단 양식 샘플 다섯 개를 시작으로 사용자 지정 모델을 학습할 수 있습니다. 첫 번째 학습 연습 후, 결과를 평가한 후 샘플을 더 추가하고 다시 학습해야 하는지 고려할 수 있습니다.

① 참고

다음 연습에서는 **사전 빌드된 영수증 모델** 만을 단계별로 진행합니다. **사용자 지정 모델**을 학습하고자 하는 경우 **Form Recognizer 설명서**를 참조하여 빠르게 시작할 수 있습니다.

Form Recognizer용 Azure 리소스

Form recognizer를 사용하려면 Azure 구독에서 Form Recognizer 리소스를 만들어야 합니다.

리소스가 만들어진 후, **키 와 앤드포인트** 를 사용하는 클라이언트 애플리케이션을 만들어 분석을 위해 제출 양식을 연결할 수 있습니다.

미리 제작된 영수증 모델 사용

현재 미리 제작된 영수증 모델은 미국에서 흔히 사용되는 일반적인 영수증을 영어로 인식하도록 설계되었습니다. 예를 들어 레스토랑, 소매점 및 주유소에서 사용되는 영수증이 있습니다. 모델은 영수증 전표에서 주요 정보를 추출할 수 있습니다.

- 거래 시간
- 거래 날짜
- 판매자 정보
- 납부한 세금
- 영수증 합계
- 영수증에 있을 수 있는 기타 관련 정보
- 영수증의 모든 텍스트도 인식되고 반환됩니다.

사용자 지정 모델을 사용할 때 최상의 결과를 얻으려면 다음 지침을 따릅니다.

- 이미지는 JPEG, PNG, BMP, PDF 또는 TIFF 형식이어야 합니다.
- 파일 크기는 50MB 미만이어야 합니다.
- 이미지의 크기는 50x50~10000x10000 픽셀 사이여야 합니다.
- PDF 문서의 경우, 17인치x17인치 이하여야 합니다.

유료 구독과 마찬가지로 영수증 모델에 대한 무료 계층 구독 계획이 있습니다. 무료 계층의 경우, PDF 또는 TIFF 형식 문서를 전달할 때 첫 200페이지만 처리됩니다.

① 참고

Form Recognizer는 이 콘텐츠가 작성된 시점의 미리 보기이며, 결과적으로 기능 및 사용량 세부 정보가 변경될 수 있습니다. 최신 정보는 서비스의 [공식 페이지](#)를 참조해야 합니다.

다음 단원: 연습 - Form Recognizer를 사용하여 영수증 분석

계속 >

< 이전

단위 3/5 ▼

다음 >

100 XP



연습 - Form Recognizer를 사용하여 영수증 분석

15분

Form Recognizer 서비스를 사용한 영수증 처리 방법을 알아보는 가장 좋은 방법은 직접 살펴보는 것입니다.

시작하기 전에

이 연습을 완료하려면 다음이 필요합니다.

- Microsoft Azure 구독. 구독이 아직 없다면 <https://azure.microsoft.com/free>에서 평가판에 가입할 수 있습니다.
- MicrosoftDocs/ai-fundamentals GitHub 리포지토리에서 Jupyter Notebook을 실행할 수 있는 Python 환경. GitHub의 랩 파일을 자신의 Python 환경(있는 경우)에 복제하거나, 아래 지침에 따라 Azure 구독에서 Azure Machine Learning 작업 영역을 만들 수 있습니다.

! 참고

이 모듈은 Azure Machine Learning 작업 영역을 사용하는 여러 가지 모듈 중 하나입니다.

Azure AI 기본 사항 인증의 준비 과정에서 이 모듈을 완료하는 경우, 작업 영역을 한 번 만들고 해당 인증 트랙의 모든 모듈에 다시 사용할 수 있습니다. 연습을 완료한 후에는 정리 지침에 따라 모듈 간 컴퓨팅 리소스를 중지해야 합니다.

Azure Machine Learning 작업 영역 만들기

Azure 구독에 Azure Machine Learning 작업 영역이 이미 있는 경우 새 브라우저 탭에서 [Azure Machine Learning Studio](#)로 이동하고 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다. 없는 경우 다음 단계에 따라 새 작업 영역을 만듭니다.

1. Azure 구독과 연결된 Microsoft 계정을 사용하여 [Azure Portal](#)에 로그인합니다.
2. + 리소스 만들기를 선택하고, *Machine Learning*을 검색한 뒤 다음 설정을 사용하여 새 *Machine Learning* 리소스를 만듭니다.
 - **작업 영역 이름:** 원하는 고유한 이름을 입력
 - **구독:** 자신의 Azure 구독
 - **리소스 그룹:** 고유한 이름의 새 리소스 그룹 만들기
 - **위치:** 사용 가능한 위치 선택

3. 작업 영역 리소스가 만들어질 때까지 기다립니다(몇 분이 걸릴 수 있음). 그런 다음 포털로 이동하고 작업 영역의 **개요** 페이지에서 Azure Machine Learning Studio를 실행(또는 새 브라우저 탭을 열고 <https://ml.azure.com> 으로 이동)한 다음 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다.
4. Azure Machine Learning Studio에서 왼쪽 상단에 있는 **☰** 아이콘을 설정/해제하면 인터페이스에 다양한 페이지가 표시됩니다. 해당 페이지를 사용하여 작업 영역에서 리소스를 관리할 수 있습니다.

컴퓨팅 인스턴스 만들기

이 연습에서 사용된 Notebook을 실행하려면 Azure Machine Learning 작업 영역에 컴퓨팅 인스턴스가 필요합니다. 컴퓨팅 인스턴스가 이미 있는 경우 시작하고, 없는 경우 다음 지침에 따라 새로 만듭니다.

1. [Azure Machine Learning Studio](#)에서 **관리** 아래에 있는 **컴퓨팅** 페이지를 확인합니다.
2. **컴퓨팅 인스턴스** 탭에서 다음 설정을 사용하여 새 컴퓨팅 인스턴스를 만듭니다.
 - **지역**: 사용 가능한 지역 선택
 - **가상 머신 유형**: CPU
 - **가상 머신 크기**: Standard_DS11_v2
 - **컴퓨팅 이름**: 고유한 이름 입력
3. 컴퓨팅 인스턴스가 시작될 때까지 기다립니다(1분 정도 걸릴 수 있음).

연습 파일 다운로드

이 모듈(및 기타 관련 모듈)에 사용되는 파일은 GitHub 리포지토리에 게시되는데, 이를 자신의 Python 환경에 복제해야 합니다. 이전 모듈에서 **ai-fundamentals** 리포지토리를 아직 복제하지 않은 경우 다음 단계를 사용하여 Azure Machine Learning 작업 영역에 복제합니다.

1. [Azure Machine Learning Studio](#)에서 **Notebooks** 페이지(**작성자** 아래)를 확인합니다. 이 페이지에는 Notebooks 실행에 사용할 수 있는 Notebook 편집기가 포함되어 있습니다.
2. **내 파일**에서 **□** 단추를 사용하여 다음 설정에 따라 새 파일을 만듭니다.
 - **파일 위치**: `Users/your user name`
 - **파일 이름**: Get-Files
 - **파일 형식**: Notebook
 - **이미 있는 경우 덮어쓰기**: 선택됨
3. 새 Notebook이 만들어지면 이전에 만든 컴퓨팅 인스턴스가 **컴퓨팅** 상자에서 선택되었는지, 그리고 상태가 **실행 중**인지 확인합니다. 그런 다음 Notebook에서 만든 사각형 셀에 다음 코드를 붙여넣습니다.

```
!git clone https://github.com/MicrosoftDocs/ai-fundamentals
```

4. 셀 옆에 있는 ▶ 단추를 사용하여 포함된 코드를 실행합니다. 이렇게 하면 GitHub에서 연습 파일이 복제됩니다.
5. 코드의 실행이 끝나고 파일의 체크아웃이 완료되면 내 파일 아래에 있는 ⌂ 단추를 사용하여 폴더 보기를 새로 고치고, **ai-fundamentals**라는 이름의 폴더가 만들어졌는지 확인합니다. 이 폴더에는 연습에 사용된 Notebook 및 기타 파일이 포함되어 있습니다.
6. Get-Files.ipynb Notebook 탭을 닫습니다.

연습 완료

Python 환경을 설정하고 **ai-fundamentals** 리포지토리를 복제했으면 Form Recognizer에서 수신 분석을 살펴볼 준비가 된 것입니다.

1. **ai-fundamentals** 폴더에서 **Receipts with Form Recognizer.ipynb** Notebook을 엽니다. Azure Machine Learning Studio에서 Notebook 편집기를 사용하는 경우, <> 단추로 파일 탐색기 창을 축소하여 Notebook 탭에 집중할 수 있는 공간을 좀 더 마련합니다.
2. Notebook의 정보를 읽고 포함된 코드 셀을 순서대로 실행합니다.

정리

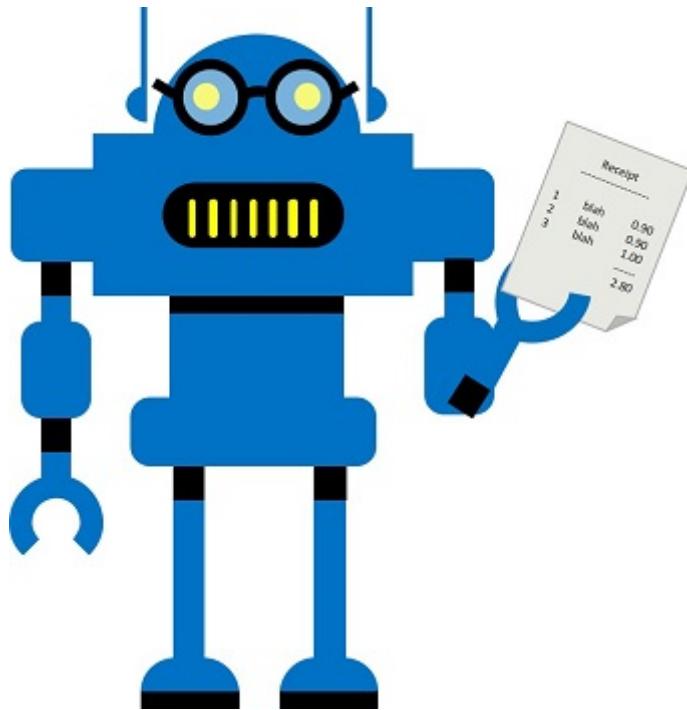
Azure Machine Learning Studio에서 컴퓨팅 인스턴스를 사용한 경우 Azure 크레딧을 불필요하게 사용하지 않도록 이를 중지해야 합니다.

1. Azure Machine Learning Studio에서 관리 아래에 있는 컴퓨팅 페이지를 확인합니다.
2. 컴퓨팅 인스턴스 탭에서 컴퓨팅 인스턴스를 선택하고 중지 단추를 사용하여 중지합니다.

다음 단원: 지식 점검

계속 >

Form Recognizer 서비스를 이용하여 영수증 분석



인공지능(AI) 컴퓨터 비전 분야에서 광학 문자 인식(OCR)은 일반적으로 인쇄된 문서나 손으로 쓴 문서를 읽는 데 사용된다. 종종, 텍스트는 추가 처리나 분석에 사용될 수 있는 형식으로 문서에서 간단히 추출된다.

In the artificial intelligence (AI) field of computer vision, optical character recognition (OCR) is commonly used to read printed or handwritten documents. Often, the text is simply extracted from the documents into a format that can be used for further processing or analysis.

보다 진보된 OCR 시나리오는 양식의 필드가 무엇을 나타내는지를 의미적으로 이해하면서 구매 주문서나 송장 같은 양식에서 정보를 추출하는 것이다. **양식 인식기** 서비스는 이러한 종류의 AI 문제를 위해 특별히 설계되었습니다.

A more advanced OCR scenario is the extraction of information from forms, such as purchase orders or invoices, with a semantic understanding of what the fields in the form represent. The **Form Recognizer** service is specifically designed for this kind of AI problem.

영수증 보기

이 예에서는 Form Recognizer'에 기본적으로 포함되어 있는 영수증 분석 모델을 사용할 것이다.

아래의 셀에 있는 **Run cell (▷)** 버튼(셀의 왼편에 있음)을 클릭하여 Form Recognizer를 이용하여 영수증을 분석하는 예제를 살펴보자.

In [1]:

```
import matplotlib.pyplot as plt
from PIL import Image
import os
%matplotlib inline

# 영수증 이미지를 로드해서 나타낸다.
fig = plt.figure(figsize=(6, 6))
image_path = os.path.join('data', 'form-receipt', 'receipt.jpg')
img = Image.open(image_path)
plt.axis('off')
plt.imshow(img)
```

Out[1]:

<matplotlib.image.AxesImage at 0x7effebbf94a8>

Northwind Traders

123 Main Street

555-123-4567

2/17/2020 13:07

1 Apple \$0.90

1 Orange \$0.80

Sub-Total \$1.70

Tax \$0.17

Total \$1.87

Form Recognizer 리소스 만들기

Azure 구독에서 Form Recognizer 리소스를 생성해보자:

1. 브라우저의 새로운 탭을 열고, Azure portal(<https://portal.azure.com>)을 (<https://portal.azure.com>)을) 입력하고, Microsoft 계정으로 로그인한다.
2. + 리소스 만들기 버튼을 클릭하고, *Form Recognizer* 서비스를 찾는다.
3. 서비스 리스트에서 **Form Recognizer**를 찾는다.
4. **Form Recognizer**에서 만들기를 클릭한다.
5. **Form Recognizer** 만들기에서 아래의 내용을 입력하고 만들기를 클릭한다.
 - **이름**: 유일한 이름을 입력한다(가능하면 영문과 숫자 사용).
 - **구독**: Azure 구독선택.
 - **위치**: 가능한 위치(한국 중부 추천):
 - **가격책정계층**: F0
 - **리소스 그룹**: 원하는 유리한 이름(가능하면 영문과 숫자 사용)
 - 검토 + 만들기를 선택한다.
6. 만들기가 완료될 때까지 기다린다.
7. Azure Portal에서 Form Recognizer 서비스가 새롭게 생성된 것을 확인하고 **Keys and Endpoint** 페이지에서 **Key1**과 **엔드포인트**값을 복사하고 아래의 셀의 **YOUR_FORM_KEY**와 **YOUR_FORM_ENDPOINT**에 붙여 넣는다.

In []:

```
form_key = 'YOUR_FORM_KEY'
form_endpoint = 'YOUR_FORM_ENDPOINT'

print('Ready to use form recognizer at {} using key {}'.format(form_endpoint, form_key))
```

Python에서 Form Recognizer 리소스를 사용하기 위해서는 Azure Form Recognizer SDK를 설치할 필요가 있다.

In []:

```
! pip install azure_ai_formrecognizer
```

영수증 분석하기

이제 Form Recognizer를 이용하여 영수증을 분석할 준비가 되었다.

In []:

```

import os
from azure.ai.formrecognizer import FormRecognizerClient
from azure.core.credentials import AzureKeyCredential

# Form Recognizer 서비스를 위한 클라이언트를 만든다.
form_recognizer_client = FormRecognizerClient(endpoint=form_endpoint, credential=AzureKeyCredential(form_key))

try:
    print("Analyzing receipt...")
    # 이미지 파일에서 영수증을 가져온다
    image_path = os.path.join('data', 'form-receipt', 'receipt.jpg')

    # 파일 데이터를 Form Recognizer에 제출한다
    with open(image_path, "rb") as f:
        analyze_receipt = form_recognizer_client.begin_recognize_receipts(receipt=f)

    # 결과 가져오기
    receipt_data = analyze_receipt.result()

    # 첫번째(그리고 유일한) 영수증으로부터 추출된 데이터를 프린트한다.
    receipt = receipt_data[0]
    receipt_type = receipt.fields.get("ReceiptType")
    if receipt_type:
        print("Receipt Type: {}".format(receipt_type.value))
    merchant_address = receipt.fields.get("MerchantAddress")
    if merchant_address:
        print("Merchant Address: {}".format(merchant_address.value))
    merchant_phone = receipt.fields.get("MerchantPhoneNumber")
    if merchant_phone:
        print("Merchant Phone: {}".format(merchant_phone.value))
    transaction_date = receipt.fields.get("TransactionDate")
    if transaction_date:
        print("Transaction Date: {}".format(transaction_date.value))
    print("Receipt items:")
    items = receipt.fields.get("Items")
    if items:
        for idx, item in enumerate(receipt.fields.get("Items").value):
            print("Wt Item #{}".format(idx+1))
            item_name = item.value.get("Name")
            if item_name:
                print("Wt - Name: {}".format(item_name.value))
            item_total_price = item.value.get("TotalPrice")
            if item_total_price:
                print("Wt - Price: {}".format(item_total_price.value))
    subtotal = receipt.fields.get("Subtotal")
    if subtotal:
        print("Subtotal: {}".format(subtotal.value))
    tax = receipt.fields.get("Tax")
    if tax:
        print("Tax: {}".format(tax.value))
    total = receipt.fields.get("Total")
    if total:
        print("Total: {}".format(total.value))

except Exception as ex:
    print('Error:', ex)

```

Form Recognizer는 양식에서 데이터를 해석하여 가맹점 주소와 전화번호, 거래 날짜와 시간, 회선 항목, 소계, 세금 및 총액을 올바르게 식별할 수 있다.

더많은 정보

Form Recognizer서비스에 대한 더 많은 정보는 다음을 참고하기 바란다. [Form Recognizer 문서](https://docs.microsoft.com/ko-kr/azure/cognitive-services/form-recognizer/index) (<https://docs.microsoft.com/ko-kr/azure/cognitive-services/form-recognizer/index>)

In []:



자연어 처리 살펴보기

▼ 2400 XP



2시간 6분 • 학습 경로 • 4 모듈

초급 AI 엔지니어 데이터 과학자 개발자 솔루션 아키텍처 학생

Cognitive Services Language Understanding Azure Portal

자연어 처리는 사용자를 보고, 듣고, 말하고, 이해할 수 있는 애플리케이션을 지원합니다. 텍스트 분석, 번역 및 언어 이해 서비스를 사용하여 Microsoft Azure를 사용하면 자연어를 지원하는 애플리케이션을 쉽게 빌드할 수 있습니다.

필수 조건

Azure Portal을 탐색할 수 있는 기능

책갈피 컬렉션에 추가

이 학습 경로의 모듈



Text Analytics 서비스를 사용한 텍스트 분석

▼ 600 XP



29분 • 모듈 • 5 단위

4.8 (677)

Text Analytics 서비스는 클라우드 기반 서비스로 감정 분석, 핵심 구 추출, 명명된 엔터티 인식 및 언어 감지를 위해서 원시 텍스트에 대한 고급 자연어 처리를 제공합니다.

개요 ▾

책갈피 컬렉션에 추가



음성 인식 및 합성

▼ 600 XP



29분 • 모듈 • 5 단위

4.8 (632)

Azure Cognitive Services를 사용하여 음성을 인식 및 합성하는 방법에 대해 알아봅니다.

개요 ▾



텍스트 및 음성 번역

✓ 600 XP



29분 • 모듈 • 5 단위

★★★★★ 4.8 (542)

AI 솔루션의 자동 번역 기능을 사용하면 언어 장벽을 허물어 보다 긴밀하게 협업할 수 있습니다.

개요 ▾



Language Understanding을 사용하여 언어 모델 만들기

✓ 600 XP



39분 • 모듈 • 5 단위

★★★★★ 4.8 (705)

이 모듈에서는 Language Understanding 서비스에 대해 소개하고, 언어를 이해하는 애플리케이션을 만드는 방법을 보여 줍니다.

개요 ▾





✓ 600 XP ➔

Text Analytics 서비스를 사용한 텍스트 분석

29분 • 모듈 • 5 단위

★★★★★ 4.8 (677)

평가하기

초급 AI 엔지니어 데이터 과학자 개발자 솔루션 아키텍처 학생

Cognitive Services

Text Analytics 서비스는 클라우드 기반 서비스로 감정 분석, 핵심 구 추출, 명명된 엔터티 인식 및 언어 감지를 위해서 원시 텍스트에 대한 고급 자연어 처리를 제공합니다.

학습 목표

텍스트 분석에 Text Analytics 서비스를 사용하는 방법을 알아봅니다.

책갈피 + 컬렉션에 추가

필수 조건

Azure Portal 탐색 능력

이 모듈은 이러한 학습 경로의 일부입니다.

[자연어 처리 살펴보기](#)

소개

3분



Azure에서 Text Analytics 시작하기

3분



연습 - Text Analytics 서비스를 사용한 텍스트 분석

20분



지식 점검

2분



요약

1분



100 XP



소개

3분

텍스트 분석은 텍스트 내용에 대한 인사이트를 얻기 위해 문서 또는 구의 다양한 측면을 평가하는 프로세스입니다. 대개 사람들은 일부 텍스트를 읽고 이면의 의미를 이해할 수 있습니다. 텍스트를 작성한 언어의 문법 규칙을 고려하지 않아도 텍스트에서 구체적인 인사이트를 파악할 수 있습니다.

예를 들어 일부 텍스트를 읽고 텍스트의 주요 논점을 나타내는 몇 가지 핵심 문구를 파악할 수 있습니다. 또한 인물 이름이나 에펠탑처럼 잘 알려진 랜드마크를 인지할 수도 있습니다. 어려울 때도 있지만, 경우에 따라서는 글쓴이가 어떤 느낌(감정)으로 텍스트를 작성했는지 느낄 수도 있습니다.

텍스트 분석 기법

텍스트 분석은 컴퓨터에서 실행되는 AI(인공 지능) 알고리즘이 텍스트에서 동일한 특성을 평가하여 구체적인 인사이트를 알아내는 프로세스입니다. 일반적으로 사람들은 인사이트를 얻기 위해 자신의 경험과 지식을 활용합니다. 컴퓨터에도 작업을 수행하려면 비슷한 정보가 제공되어야 합니다. 다음을 포함하여 텍스트를 분석하는 소프트웨어를 작성하는 데 사용할 수 있는 자주 사용되는 몇 가지 기술이 있습니다.

- 텍스트에 사용된 용어의 통계 분석입니다. 예를 들어 일반적인 “중지 단어”(텍스트에 대한 의미론적 정보를 거의 나타내지 않는 “the” 또는 “a”와 같은 단어)를 제거하고 나머지 단어의 ‘빈도 분석’(각 단어가 나오는 횟수 세기)을 수행하면 텍스트의 주요 주제에 대한 단서를 제공할 수 있습니다.
- 빈도 분석을 일반적으로 ‘N-그램’(두 단어 구는 ‘바이-그램’, 세 단어 구는 ‘트라이-그램’ 등)이라고 부르는 다중 용어구 분석으로 확장합니다.
- ‘형태소 분석’ 또는 ‘기본형 분석’ 알고리즘을 적용하여 단어를 계산하기 전에 단어를 표준화합니다. 예를 들어, “power”, “powered” 및 “powerful” 등의 단어는 동일한 단어로 해석됩니다.
- 언어적 구조 규칙을 적용하여 문장을 분석합니다. 예를 들어, ‘명사’, ‘동사’, ‘형용사’ 등을 포함하는 ‘명사구’와 같은 트리 형태의 구조로 문장을 나눕니다.
- 기계 학습 모델을 학습하는 데 사용할 수 있는 수치로 단어 또는 용어를 인코딩합니다. 예를 들어 포함된 용어를 기준으로 텍스트 문서를 분류합니다. 이 기술은 문서를 긍정적 또는 부정적으로 분류하는 ‘감정 분석’을 수행하는 데 자주 사용됩니다.
- 단어를 n차원 공간의 위치에 할당하여 단어 간의 의미론적 관계를 포착하는 ‘벡터화’ 모델을 만듭니다. 예를 들어 이 모델링 기술은 “꽃”과 “식물”이라는 단어에는 서로 가까이 위치

하도록 하는 값을 할당하고, “스케이트보드”에는 훨씬 더 멀리 위치하도록 하는 값을 줄 수 있습니다.

이러한 기술은 뛰어난 효과를 발휘할 수 있지만 프로그래밍은 복잡할 수 있습니다. Microsoft Azure에서 **Text Analytics** 인지 서비스는 다음을 수행할 수 있는 미리 학습된 모델을 사용하여 애플리케이션 개발을 간소화하는 데 도움이 될 수 있습니다.

- 문서 또는 텍스트의 언어를 판단합니다(예: 프랑스어 또는 영어).
- 텍스트에 대한 감정 분석을 수행하여 긍정적 또는 부정적 감정을 판단합니다.
- 텍스트에서 주요 논점을 나타낼 수 있는 핵심 구를 추출합니다.
- 텍스트에서 엔터티를 식별하고 분류합니다. 엔터티는 사람, 장소, 조직 또는 일상적인 항목(예: 날짜, 시간, 수량 등)일 수 있습니다.

이 모듈에서는 이러한 기능 중 일부를 살펴보고 다음과 같은 애플리케이션에 해당 기능을 적용하는 방법을 이해할 수 있습니다.

- 정치적 캠페인이나 시장에서 제품을 둘러싼 감정을 검색하는 소셜 미디어 피드 분석기
- 카탈로그에 있는 문서의 주요 주제 요약을 지원하기 위해 핵심 문구를 추출하는 문서 검색 애플리케이션
- 식별을 위해 문서 또는 다른 텍스트에서 브랜드 정보나 회사 이름을 추출하는 도구

위의 예시는 Text Analytics가 도움이 될 수 있는 여러 영역 중 일부입니다.

다음 단원: Azure에서 Text Analytics 시작하기

계속 >

< 이전

단위 2/5 ▼

다음 >

✓ 100 XP



Azure에서 Text Analytics 시작하기

3분

Text Analytics 서비스는 원시 텍스트를 통해 고급 자연어 처리를 수행할 수 있는 Azure Cognitive Services 제공 서비스의 일부입니다.

Text Analytics 서비스용 Azure 리소스

애플리케이션에서 Text Analytics 서비스를 사용하려면 Azure 구독에 적절한 리소스를 프로비저닝해야 합니다. 다음 종류의 리소스 중 하나를 프로비저닝하도록 선택할 수 있습니다.

- **Text Analytics** 리소스 - Text Analytics 서비스를 사용하려는 경우 또는 다른 서비스와 별도로 리소스에 대한 액세스 및 청구를 관리하려는 경우, 이 리소스 유형을 선택합니다.
- **Cognitive Services** 리소스 - 다른 인지 서비스와 함께 Text Analytics 서비스를 사용하려는 경우 이 리소스 종류를 선택하고, 이 서비스에 대한 액세스 및 청구를 함께 관리하려고 합니다.

언어 감지

Text Analytics 서비스의 언어 감지 기능을 사용하여 텍스트가 작성된 언어를 식별합니다. 분석을 위해 한 번에 여러 문서를 제출할 수 있습니다. 제출된 각 문서에 대해 서비스는 다음을 검색합니다.

- 언어 이름(예: “영어”)
- ISO 6391 언어 코드(예: “en”)
- 언어 감지에 대한 신뢰 수준을 나타내는 점수

예를 들어 고객이 설문조사를 완료하고 음식, 서비스, 직원 등에 대한 피드백을 제공할 수 있는 레스토랑을 소유하고 운영하는 시나리오를 생각해 보겠습니다. 고객으로부터 다음과 같은 리뷰를 받았다고 가정해 보세요.

리뷰 1 : “A fantastic place for lunch. The soup was delicious.”

리뷰 2 : “Comida maravillosa y gran servicio.”

리뷰 3 : “The croque monsieur avec frites was terrific. Bon appetit!”

Text Analytics 서비스를 사용하여 이러한 각 리뷰에 대한 언어를 검색할 수 있고, 서비스에서 다음과 같은 결과로 응답할 수 있습니다.

문서	언어 이름	ISO 6391 코드	점수
리뷰 1	영어	en	1.0
리뷰 2	스페인어	es	1.0
리뷰 3	영어	en	0.9

영어와 프랑스어가 혼합된 텍스트이지만 리뷰 3에서 감지된 언어는 영어입니다. 언어 감지 서비스는 텍스트에서 **지배적** 언어에 중점을 둡니다.* 이 서비스는 알고리즘을 사용하여 텍스트의 다른 언어와 비교하고 언어에 대한 구길이 또는 총 텍스트 양과 같은 지배적 언어를 결정합니다. 지배적 언어는 언어 코드와 함께 값이 반환됩니다. 혼합 언어 텍스트의 결과로 신뢰도 점수가 1 미만일 수 있습니다.

모호하거나 혼합된 언어 콘텐츠

속성이 모호하거나 언어 콘텐츠가 혼합된 텍스트가 있을 수 있습니다. 이러한 상황은 서비스에 문제를 일으킬 수 있습니다. 모호한 콘텐츠 예시는 문서에 제한된 텍스트 또는 문장 부호만 포함하는 경우입니다. 예를 들어 서비스를 사용하여 “:-)”라는 텍스트를 분석하면 언어 이름과 언어 식별자 값이 알 수 없음 *이고, 점수는 ‘숫자가 아님’을 뜻하는 NaN 으로 표시됩니다.

감정 분석

Text Analytics 서비스는 텍스트를 평가하고 각 문장에 대한 감정 점수 및 레이블을 반환할 수 있습니다. 이 기능은 소셜 미디어, 고객 리뷰, 토론 포럼 등에서 긍정적이고 부정적인 감정을 감지하는 데 유용합니다.

이 서비스는 미리 빌드된 기계 학습 분류 모델을 사용하여 텍스트를 평가하고 0에서 1 범위의 감정 점수를 반환하는데, 1에 가까운 값일수록 긍정적인 감정을 나타냅니다. 범위의 중간(0.5)에 가까운 점수는 중립 또는 불확실로 간주합니다.

예를 들어, 다음 레스토랑 리뷰 두 개의 감정을 분석할 수 있습니다.

“어젯밤 이 레스토랑에서 저녁 식사를 했으며 직원들의 정중한 태도가 첫눈에 들어왔습니다. 직원들은 우리에게 친절한 태도로 인사를 하고 테이블로 곧장 안내했습니다. 테이블은 깨끗했고, 의자는 편안했으며, 음식은 정말 맛있었습니다.”

그리고

“이 레스토랑에서 경험한 저녁 식사는 최악 중 하나였습니다. 서비스가 느렸고 음식은 형편 없었습니다. 다시는 여기에서 식사하지 않을 것입니다.”

첫 번째 리뷰에 대한 감정 점수는 약 0.9일 수 있으며, 이는 긍정적인 감정을 나타냅니다. 한편 두 번째 리뷰 점수는 0.1에 가까울 수 있으며, 이는 부정적인 감정을 나타냅니다.

불확실한 감정

점수 0.5는 텍스트의 감정이 불확실함을 나타낼 수 있으며, 이는 감정을 드러낼 만한 충분한 맥락이 없거나 불충분한 구문의 텍스트로 인해 발생할 수 있습니다. 예를 들어 구조가 없는 문장의 단어 목록은 불확실한 점수를 유발할 수 있습니다. 점수가 0.5일 수 있는 또 다른 예는 잘못된 언어 코드를 사용한 경우입니다. 언어 코드(예: 영어의 경우 "en", 프랑스어의 경우 "fr")는 텍스트에 사용된 언어를 분석 서비스에 알리는 데 사용됩니다. 프랑스어로 텍스트를 전달하지만 서비스에 언어 코드를 영어인 `en`이라고 알려준다면 서비스는 정확하게 0.5의 점수를 반환합니다.

핵심 구 추출

핵심 구 추출은 문서 한 개 또는 여러 개의 텍스트를 평가한 다음 문서의 주요 논점을 파악하는 개념입니다. 앞서 설명한 레스토랑 시나리오를 생각해보겠습니다. 수집한 설문조사의 양에 따라 리뷰를 전부 읽는 데 시간이 오래 걸릴 수 있습니다. 그 대신 Text Analytics 서비스의 핵심 구 추출 기능을 사용해서 주요 논점을 요약할 수 있습니다.

다음과 같은 리뷰를 받을 수 있습니다.

“여기서 생일 축하를 위해 저녁 식사를 했고 환상적인 경험이었습니다. 친절한 여주인이 우리에게 인사를 하고 바로 테이블로 데려갔습니다. 분위기는 편안했고, 음식은 훌륭했으며, 서비스 또한 뛰어났습니다. 훌륭한 음식과 세심한 서비스를 원한다면 이 장소를 추천합니다.”

핵심 구 추출은 다음과 같은 구를 추출하여 이 리뷰에 몇 가지 컨텍스트를 제공할 수 있습니다.

- 세심한 서비스
- 훌륭한 음식
- 생일 축하
- 환상적인 경험
- 테이블
- 친절한 여주인
- 저녁 식사
- 분위기
- 장소

감정 분석을 사용하여 이 리뷰가 긍정적이라고 판단할 수 있을 뿐만 아니라, 핵심 구를 사용하여 리뷰의 중요한 요소를 식별할 수 있습니다.

엔터티 인식

Text Analytics 서비스에 구조화되지 않은 텍스트를 제공하면, 인식한 텍스트의 '엔터티' 목록을 반환합니다. 이 서비스는 웹상의 해당 엔터티에 대한 보다 자세한 정보로 연결하는 링크를 제공할 수도 있습니다. 엔터티는 기본적으로 특정 유형 또는 범주의 항목입니다. 경우에 따라 다음 표에 나온 것과 같은 하위 유형이 있습니다.

유형	하위 유형	예
사람		"빌 게이츠", "존"
위치		"파리", "뉴욕"
조직		"Microsoft"
수량	숫자	"6" 또는 "여섯"
수량	백분율	"25%" 또는 "오십 퍼센트"
수량	서수	"1st" 또는 "첫 번째"
수량	나이	"생후 90일" 또는 "30세"
수량	통화	"10.99"
수량	차원	"10마일", "40cm"
수량	온도	"45도"
DateTime		"2012년 2월 4일, 오후 6:30"
DateTime	날짜	"2017년 5월 2일" 또는 "2017/05/02"
DateTime	시간	"8am" 또는 "8:00"
DateTime	DateRange	"5월 2일~5월 5일"
DateTime	TimeRange	"오후 6시부터 오후 7시"
DateTime	기간	"1분 45초"
DateTime	설정	"매주 화요일"
URL		" https://www.bing.com "

유형	하위 유형	예
메일		"support@microsoft.com"
미국 국내 전화 번호		"(312) 555-0176"
IP 주소		"10.0.1.125"

또한 이 서비스는 특정 참조에 연결하여 엔터티를 구분하도록 도와주는 '엔터티 링크' 설정을 지원합니다. 인식된 엔터티에 대해 서비스는 관련 'Wikipedia' 아티클 URL을 반환합니다.

예를 들어, Text Analytics 서비스를 사용하여 다음 레스토랑 리뷰 추출의 엔터티를 검색한다고 가정해 보겠습니다.

“지난주에 시애틀에 있는 레스토랑에서 식사했습니다.”

엔터티	유형	하위 유형	Wikipedia URL
시애틀	위치		https://en.wikipedia.org/wiki/Seattle
지난주	DateTime	DateRange	

다음 단원: 연습 - Text Analytics 서비스를 사용한 텍스트 분석

계속 >

< 이전

단위 3/5 ▼

다음 >

✓ 100 XP

연습 - Text Analytics 서비스를 사용한 텍스트 분석

20분

Text Analysis 서비스를 사용하여 텍스트를 분석하는 방법을 배우는 가장 좋은 방법은 스스로 알아보는 것입니다.

시작하기 전에

이 연습을 완료하려면 다음이 필요합니다.

- Microsoft Azure 구독. 구독이 아직 없다면 <https://azure.microsoft.com/free>에서 평가판에 가입할 수 있습니다.
- MicrosoftDocs/ai-fundamentals GitHub 리포지토리에서 Jupyter Notebook을 실행할 수 있는 Python 환경. GitHub의 랩 파일을 자신의 Python 환경(있는 경우)에 복제하거나, 아래 지침에 따라 Azure 구독에서 Azure Machine Learning 작업 영역을 만들 수 있습니다.

① 참고

이 모듈은 Azure Machine Learning 작업 영역을 사용하는 여러 가지 모듈 중 하나입니다.

Azure AI 기본 사항 인증의 준비 과정에서 이 모듈을 완료하는 경우, 작업 영역을 한 번 만들고 해당 인증 트랙의 모든 모듈에 다시 사용할 수 있습니다. 연습을 완료한 후에는 정리 지침에 따라 모듈 간 컴퓨팅 리소스를 중지해야 합니다.

Azure Machine Learning 작업 영역 만들기

Azure 구독에 Azure Machine Learning 작업 영역이 이미 있는 경우 새 브라우저 탭에서 [Azure Machine Learning Studio](#)로 이동하고 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다. 없는 경우 다음 단계에 따라 새 작업 영역을 만듭니다.

1. Azure 구독과 연결된 Microsoft 계정을 사용하여 [Azure Portal](#)에 로그인합니다.
2. + 리소스 만들기를 선택하고, *Machine Learning*을 검색한 뒤 다음 설정을 사용하여 새 Machine Learning 리소스를 만듭니다.
 - **작업 영역 이름:** 원하는 고유한 이름을 입력
 - **구독:** 자신의 Azure 구독
 - **리소스 그룹:** 고유한 이름의 새 리소스 그룹 만들기

- **위치:** 사용 가능한 위치 선택

3. 작업 영역 리소스가 만들어질 때까지 기다립니다(몇 분이 걸릴 수 있음). 그런 다음 포털로 이동하고 작업 영역의 **개요** 페이지에서 Azure Machine Learning Studio를 실행(또는 새 브라우저 탭을 열고 <https://ml.azure.com> 으로 이동)한 다음 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다.
4. Azure Machine Learning Studio에서 왼쪽 상단에 있는 ⓢ 아이콘을 설정/해제하면 인터페이스에 다양한 페이지가 표시됩니다. 해당 페이지를 사용하여 작업 영역에서 리소스를 관리할 수 있습니다.

컴퓨팅 인스턴스 만들기

이 연습에서 사용된 Notebook을 실행하려면 Azure Machine Learning 작업 영역에 컴퓨팅 인스턴스가 필요합니다. 컴퓨팅 인스턴스가 이미 있는 경우 시작하고, 없는 경우 다음 지침에 따라 새로 만듭니다.

1. [Azure Machine Learning Studio](#) 에서 **관리** 아래에 있는 **컴퓨팅** 페이지를 확인합니다.
2. **컴퓨팅 인스턴스** 탭에서 다음 설정을 사용하여 새 컴퓨팅 인스턴스를 만듭니다.
 - **지역:** 사용 가능한 지역 선택
 - **가상 머신 유형:** CPU
 - **가상 머신 크기:** Standard_DS11_v2
 - **컴퓨팅 이름:** 고유한 이름 입력
3. 컴퓨팅 인스턴스가 시작될 때까지 기다립니다(1분 정도 걸릴 수 있음).

연습 파일 다운로드

이 모듈(및 기타 관련 모듈)에 사용되는 파일은 GitHub 리포지토리에 게시되는데, 이를 자신의 Python 환경에 복제해야 합니다. 이전 모듈에서 **ai-fundamentals** 리포지토리를 아직 복제하지 않은 경우 다음 단계를 사용하여 Azure Machine Learning 작업 영역에 복제합니다.

1. [Azure Machine Learning Studio](#) 에서 **Notebooks** 페이지(**작성자** 아래)를 확인합니다. 이 페이지에는 Notebook 실행에 사용할 수 있는 Notebook 편집기가 포함되어 있습니다.
2. **내 파일**에서 □ 단추를 사용하여 다음 설정에 따라 새 파일을 만듭니다.
 - **파일 위치:** Users/your user name
 - **파일 이름:** Get-Files
 - **파일 형식:** Notebook
 - **이미 있는 경우 덮어쓰기:** 선택됨
3. 새 Notebook이 만들어지면 이전에 만든 컴퓨팅 인스턴스가 **컴퓨팅** 상자에서 선택되었는지, 그리고 상태가 **실행 중**인지 확인합니다. 그런 다음 Notebook에서 만든 사각형 셀에

다음 코드를 붙여넣습니다.

복사

```
!git clone https://github.com/MicrosoftDocs/ai-fundamentals
```

4. 셀 옆에 있는 ▶ 단추를 사용하여 포함된 코드를 실행합니다. 이렇게 하면 GitHub에서 연습 파일이 복제됩니다.
5. 코드의 실행이 끝나고 파일의 체크아웃이 완료되면 내 파일 아래에 있는 ⌂ 단추를 사용하여 폴더 보기 를 새로 고치고, **ai-fundamentals**라는 이름의 폴더가 만들어졌는지 확인합니다. 이 폴더에는 연습에 사용된 Notebook 및 기타 파일이 포함되어 있습니다.
6. Get-Files.ipynb Notebook 탭을 닫습니다.

연습 완료

Python 환경을 설정하고 **ai-fundamentals** 리포지토리를 복제했으면 텍스트 분석을 살펴볼 준비가 된 것입니다.

1. **ai-fundamentals** 폴더에서 **Text Analytics.ipynb** Notebook을 엽니다. Azure Machine Learning Studio에서 Notebook 편집기를 사용하는 경우, <> 단추로 파일 탐색기 창을 축소하여 Notebook 탭에 집중할 수 있는 공간을 좀 더 마련합니다.
2. Notebook의 정보를 읽고 포함된 코드 셀을 순서대로 실행합니다.

정리

Azure Machine Learning Studio에서 컴퓨팅 인스턴스를 사용한 경우 Azure 크레딧을 불필요하게 사용하지 않도록 이를 중지해야 합니다.

1. Azure Machine Learning Studio에서 관리 아래에 있는 컴퓨팅 페이지를 확인합니다.
2. 컴퓨팅 인스턴스 탭에서 컴퓨팅 인스턴스를 선택하고 중지 단추를 사용하여 중지합니다.

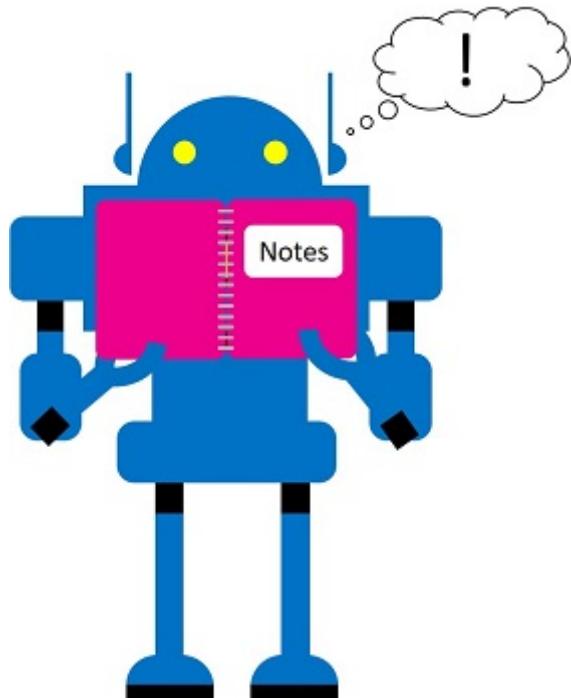
다음 단원: 지식 점검

계속 >

Text Analytics

자연어 처리(Natural Language Processing, NLP)는 문어와 구어를 다루는 인공지능(AI)의 한 분야이다. NLP를 사용하여 텍스트나 음성에서 의미적 의미를 추출하거나 자연 언어로 의미 있는 응답을 공식화하는 솔루션을 구축할 수 있다.

Microsoft Azure Cognitive 서비스에는 *Text Analytics* 서비스가 포함되어 있다. 이 서비스는 텍스트의 주요 구문(Key Phrases)을 식별하고 감정에 따라 텍스트를 분류하는 등 기본적으로 제공되는 NLP 기능을 포함한다.



예를 들어 가상의 *Margie's Travel* 조직이 고객에게 호텔 숙박에 대한 리뷰를 제출하도록 요청했다고 가정해보겠다. *Text Analytics* 서비스를 사용하여 주요 구문을 추출하여 리뷰를 요약하고, 어떤 리뷰가 긍정적이고 어떤 리뷰가 부정적인지 확인하거나, 또는 위치나 사람과 같은 알려진 엔티티를 추출하는 등 리뷰 텍스트를 분석할 수 있다.

리뷰 문서 보기

먼저 고객이 남긴 호텔 리뷰를 살펴보자.

리뷰는 텍스트 파일이다. 그 내용들을 보기 위해서는 아래의 셀 왼편에 있는 **Run cell (▷)** 버튼을 클릭하면 된다.

In []:

```
import os

# /data/reviews 폴더에 있는 리뷰들을 읽는다.
reviews_folder = os.path.join('data', 'text', 'reviews')

# ID(파일 이름)과 텍스트 (내용)속성들로 구성된 리뷰들의 컬렉션을 생성한다.
reviews = []
for file_name in os.listdir(reviews_folder):
    review_text = open(os.path.join(reviews_folder, file_name)).read()
    review = {"id": file_name, "text": review_text}
    reviews.append(review)

for review_num in range(len(reviews)):
    # 리뷰 텍스트를 인쇄한다
    print('{}\n{}'.format(reviews[review_num]['id'], reviews[review_num]['text']))
```

Cognitive 서비스 리소스 만들기

이러한 리뷰의 텍스트를 분석하려면 **Text Analytics** 인지 서비스를 사용하면 된다. 이를 사용하려면 Azure 구독에 **Text Analytics*** 또는 **Cognitive Services***** 리소스를 프로비저닝해야 한다.(이 서비스가 사용하려는 유일한 서비스이거나 사용을 별도로 추적하려는 경우 **Text Analytics** 리소스를 사용하면 된다. 그렇지 않으면 **Cognitive Services** 리소스를 사용하여 **Text Analytics** 서비스를 다른 **Cognitive** 서비스와 결합할 수 있다. 이렇게 하면 개발자는 하나의 앤드포인트와 키를 사용하여 액세스할 수 있다.)

이미 만들어진 Cognitive 서비스가 없다면 다음과 같은 순서로 Azure 구독에서 **Cognitive Services** 리소스를 생성할 수 있다.

1. 브라우저의 새로운 탭을 열고, Azure portal(<https://portal.azure.com>)을 (<https://portal.azure.com>)을 입력하고, Microsoft 계정으로 로그인한다..
2. + 리소스 만들기 버튼을 클릭하고, **Cognitive Services** 서비스를 찾은 다음, **Cognitive Services** 리소스를 다음과 같은 내용으로 생성한다.
 - **이름**: 유일한 이름을 입력한다(가능하면 영문과 숫자사용).
 - **구독**: Azure 구독선택.
 - **위치**: 가능한 위치(한국 중부 추천):
 - **가격책정계층**: 표준 S0
 - **리소스 그룹**: 원하는 유리한 이름(가능하면 영문과 숫자사용).
3. 배포가 완료될 때까지 기다린다. 그런 다음 Cognitive Services 리소스로 이동하여 *개요 페이지에서 링크를 클릭하여 서비스 키를 관리한다. 클라이언트 응용 프로그램에서 Cognitive Services 리소스에 연결하려면 앤드포인트와 키가 필요하다.

Cognitive Services 리소스에 있는 키와 앤드포인트 가져오기

Cognitive Services 리소스를 사용하기 위해서는, 클라이언트 응용프로그램에서는 앤드포인트와 인증 키가 필요하다.

1. Azure portal에서, Cognitive Services 리소스를 선택하고 **키 및 앤트포인트** 페이지를 선택한 다음 **키1** 을 복사하여 아래의 **YOUR_COG_KEY**.를 붙여 넣는다.
2. 리소스에 있는 **엔드포인트** 를 복사해서 아래의 **YOUR_COG_ENDPOINT**.에 붙여 넣는다.
3. 셀을 선택한 다음 셀 왼쪽에있는 **셀 실행(▷)** 버튼을 클릭하여 아래 코드를 실행한다.

In []:

```
cog_key = 'YOUR_COG_KEY'
cog_endpoint = 'YOUR_COG_ENDPOINT'

print('Ready to use cognitive services at {} using key {}'.format(cog_endpoint, cog_key))
```

만들어진 Cognitive 서비스 리소스에서 Text Analytics 서비스를 사용하기 위해서는 Azure Cognitive Services Text Analytics SDK를 설치해야 한다.

In []:

```
! pip install azure-cognitiveservices-language-textanalytics
```

언어 검지하기

리뷰가 쓰여진 언어를 확인하는 작업부터 시작해보자.

In []:

```
import os
from azure.cognitiveservices.language.textanalytics import TextAnalyticsClient
from msrest.authentication import CognitiveServicesCredentials

# Text Analytics Cognitive 서비스 리소스를 위한 클라이언트 가져오기
text_analytics_client = TextAnalyticsClient(endpoint=cog_endpoint,
                                             credentials=CognitiveServicesCredentials(cog_key))

# 앞에서 언급한 /data/reviews 폴더로부터 읽은 리뷰들을 분석한다
language_analysis = text_analytics_client.detect_language(documents=reviews)

# 각 리뷰에서 감지된 언어 세부 정보들을 프린트한다.
for review_num in range(len(reviews)):
    # print the review id
    print(reviews[review_num]['id'])

    # 이 리뷰에 대한 언어 세부 정보를 가져온다.
    lang = language_analysis.documents[review_num].detected_languages[0]
    print(' - Language: {} \n - Code: {} \n - Score: {}'.format(lang.name, lang.iso6391_name,
                                                               lang.score))

    # 리뷰 컬렉션에 검지된 언어 코드를 추가한다(이렇게 함으로 더 많은 분석이 가능하다).
    reviews[review_num]["language"] = lang.iso6391_name
```

핵심 구(Key Phrases) 추출하기

이제 고객 리뷰의 텍스트를 분석하여 주요 논점을 나타내는 핵심 구를 식별할 수 있다.

In []:

```
# 이전 코드 셀에서 생성한 클라이언트와 리뷰들을 이용하여 핵심구를 추출한다
key_phrase_analysis = text_analytics_client.key_phrases(documents=reviews)

# 각 리뷰별로 핵심 구를 인쇄한다
for review_num in range(len(reviews)):
    # 리뷰 ID를 인쇄한다.
    print(reviews[review_num]['id'])

    # 이 리뷰에 있는 핵심 구를 가져온다.
    print('WnKey Phrases:')
    key_phrases = key_phrase_analysis.documents[review_num].key_phrases
    # 각 핵심 구를 인쇄한다.
    for key_phrase in key_phrases:
        print('Wt', key_phrase)
    print('Wn')
```

핵심 구는 리뷰 단위로 가장 중요한 논점을 이해하는 데 도움이 될 수 있다. 예를 들어, "도움이 되는 직원(helpful staff)" 또는 "서비스 불량(poor service)"이라는 문구가 포함된 리뷰는 주요 관심사를 나타낼 수 있다.

감정분석(정서분석)

감정 점수를 기준으로 리뷰를 긍정적 또는 부정적으로 분류하는 것이 유용할 수 있다. 이것도 Text Analytics 서비스를 사용하여 수행할 수 있다.

In []:

```
# 이전 코드 셀에서 생성한 클라이언트와 리뷰들을 이용하여 감정 점수를 가져온다
sentiment_analysis = text_analytics_client.sentiment(documents=reviews)

# 리뷰들에 대한 결과를 프린트한다.
for review_num in range(len(reviews)):

    # 이 리뷰에 대한 감정점수를 가져온다.
    sentiment_score = sentiment_analysis.documents[review_num].score

    # 0.5보기 크면 '긍정적'으로 분류한다.
    if sentiment_score < 0.5:
        sentiment = 'negative'
    else:
        sentiment = 'positive'

    # 파일 이름과 감정을 프린트한다.
    print('{} : {} ({})'.format(reviews[review_num]['id'], sentiment, sentiment_score))
```

알려진 엔터티 추출하기

*Entities*는 일반적으로 널리 이해되는 유형의 항목들을 의미하며 텍스트안에 언급되어 있는 것들의 목록이다. 예를 들면 장소, 사람 또는 날짜 등이다. 리뷰에서 언급되어 있는 날짜와 장소에 관심이 있다고 가정하면 다음 코드를 실행하여 그런 것들을 찾을 수 있다.

In []:

```
# 이전 코드 셀에서 생성한 클라이언트와 리뷰들을 이용하여 정한 엔터티들을 가져온다
entity_analysis = text_analytics_client.entities(documents=reviews)

# 리뷰들에 대한 결과들을 프린트한다
for review_num in range(len(reviews)):
    print(reviews[review_num]['id'])
    # 이 리뷰에 포함되어 있는 엔터티들을 가져온다
    entities = entity_analysis.documents[review_num].entities
    for entity in entities:
        # 장소 날짜와 엔터티들을 가져온다
        if entity.type in ['DateTime', 'Location']:
            link = '(' + entity.wikipedia_url + ')' if entity.wikipedia_id is not None else ''
            print(' - {}: {} {}'.format(entity.type, entity.name, link))
```

일부 엔티티는 연결된 Wikipedia 페이지가 있을 정도로 잘 알려져 있으며, 이 경우 Text Analytics 서비스는 해당 페이지의 URL을 반환한다.

심화학습

Text Analytics 서비스에 대한 더많은 정보는 다음을 참조하기 바란다. [Text Analytics service documentation \(https://docs.microsoft.com/azure/cognitive-services/text-analytics/\)](https://docs.microsoft.com/azure/cognitive-services/text-analytics/)



✓ 600 XP ➔

음성 인식 및 합성

29분 • 모듈 • 5 단위

★★★★★ 4.8 (632)

[평가하기](#)[중급](#) [AI 엔지니어](#) [개발자](#) [솔루션 아키텍처](#) [학생](#) [Azure](#) [Cognitive Services](#)[SDK](#)

Azure Cognitive Services를 사용하여 음성을 인식 및 합성하는 방법에 대해 알아봅니다.

학습 목표

이 모듈에서 학습할 내용은 다음과 같습니다.

- 음성 인식 및 합성에 대해 알아보기
- Azure에서 음성 인식 서비스를 사용하는 방법 알아보기

 [책갈피](#) [컬렉션에 추가](#)

필수 조건

Azure Portal 탐색 능력

이 모듈은 이러한 학습 경로의 일부입니다.

[자연어 처리 살펴보기](#)

소개

3분



Azure에서 음성 시작

3분



연습 - 음성 서비스 사용

20분



지식 점검

2분



요약

1분



✓ 100 XP



소개

3분

인공 지능(AI) 솔루션이 구두 명령을 받아들이고 음성 응답을 제공하는 서비스가 더욱 늘어날 것으로 기대됩니다. “조명 끄기”와 같은 명령을 내리고 “오늘 비가 올 것인가?”와 같은 질문에 구두 답변을 요청하는 등 말로 통제할 수 있는 홈 시스템 및 자동차 시스템이 더욱 증가하게 될 것입니다.

이러한 종류의 상호 작용이 가능하려면 AI 시스템이 다음 두 가지 기능을 지원해야 합니다.

- **음성 인식** - 음성 입력을 감지하고 해석하는 기능.
- **음성 합성** - 음성 출력을 생성하는 기능.

음성 인식

음성 인식은 말해진 단어를 텍스트 표현으로 전사하는 방법을 통해 처리할 수 있는 데이터로 변환하는 것과 관련이 있습니다. 말해진 단어는 오디오 파일의 녹음된 음성 또는 마이크에서 나오는 라이브 오디오 형식이 될 수 있습니다. 음성 패턴은 오디오에서 분석되어 단어에 매핑되는 인식 가능한 패턴을 결정합니다. 이를 위해 소프트웨어는 일반적으로 다음을 비롯한 여러 유형의 모델을 사용합니다.

- 오디오 신호를 음소(특정 사운드를 나타내는 단위)로 변환하는 ‘음향’ 모델.
- 음소를 단어로 매핑하는 ‘언어’ 모델(일반적으로 음소에 따라 가장 가능성성이 높은 단어 시퀀스를 예측하는 통계 알고리즘 사용).

인식된 단어는 일반적으로 다음과 같이 다양한 목적에 사용할 수 있는 텍스트로 변환됩니다.

- 녹화된 동영상 또는 라이브 비디오에 대한 자막 제공
- 전화 통화 또는 회의 내용 대본 만들기
- 자동화된 메모 받아쓰기
- 추가 처리를 위해 의도한 사용자 입력 결정

음성 합성

음성 합성은 대부분 음성 인식의 반대입니다. 음성 합성은 일반적으로 텍스트를 음성으로 변환하여 데이터를 음성화하는 것과 관련이 있습니다. 음성 합성 솔루션에는 다음 정보가 필요합니다.

- 읽을 텍스트.

- 말을 음성화하는 데 사용할 음성.

음성을 합성하기 위해 시스템은 일반적으로 텍스트를 '토큰화'하여 개별 단어로 분할하고 각 단어에 음성 발음을 할당합니다. 그런 다음 오디오 형식으로 변환될 음소를 만들기 위해 음성 전사를 '운율' 단위(예: 구, 절 또는 문장)로 세분화합니다. 그런 다음 음고 및 음색과 같은 매개 변수를 결정하는 음성을 적용하고 스피커로 출력되거나 파일에 기록될 수 있는 오디오 파형을 생성하여 이러한 음소를 오디오로 합성합니다.

다음과 같이 다양한 목적으로 음성 합성의 출력을 사용할 수 있습니다.

- 사용자 입력에 대한 음성 응답 생성.
- 전화 시스템용 음성 메뉴 만들기.
- 핸즈프리 시나리오에서 메일 또는 문자 메시지를 소리 내어 읽기.
- 기차역 또는 공항과 같은 공공장소에서 공지 사항 방송.

다음 단원: Azure에서 음성 시작

[계속 >](#)

< 이전

단위 2/5 ▼

다음 >

✓ 100 XP



Azure에서 음성 시작

3분

Microsoft Azure는 다음과 같은 API(애플리케이션 프로그래밍 인터페이스)를 포함하는 음성 인지 서비스를 통해 음성 인식 및 음성 합성을 모두 제공합니다.

- **Speech-to-Text API**
- **Text-to-Speech API**

음성 서비스용 Azure 리소스

애플리케이션에서 음성 서비스를 사용하려면 Azure 구독에 적절한 리소스를 프로비저닝해야 합니다. 다음 종류의 리소스 중 하나를 프로비저닝하도록 선택할 수 있습니다.

- **음성 리소스** - 음성 서비스를 사용하려는 경우 또는 다른 서비스와 별도로 리소스에 대한 액세스 및 청구를 관리하려는 경우 이 리소스 종류를 선택합니다.
- **Cognitive Services** 리소스 - 다른 인지 서비스와 함께 음성 서비스를 사용할 계획이고 이러한 서비스에 대한 액세스 및 청구를 함께 관리하려는 경우 이 리소스 종류를 선택합니다.

Speech to Text API

Speech to Text API를 사용하여 오디오의 실시간 또는 일괄 전사를 텍스트 형식으로 수행할 수 있습니다. 전사용 오디오 소스는 마이크 또는 오디오 파일에서 나오는 실시간 오디오 스트림일 수 있습니다.

Speech to Text API에서 사용하는 모델은 Microsoft에서 학습한 범용 언어 모델을 기반으로 합니다. 모델에 사용하는 데이터는 Microsoft 소유이며 Microsoft Azure에 배포됩니다. 이 모델은 대화 및 받아쓰기라는 두 가지 시나리오에 최적화되어 있습니다. Microsoft에서 미리 빌드된 모델이 필요한 항목을 제공하지 않는 경우 음향, 언어 및 발음을 포함한 사용자 지정 모델을 만들고 학습할 수도 있습니다.

실시간 전사

실시간 Speech to Text를 이용하여 오디오 스트림에서 텍스트를 전사할 수 있습니다. 프레젠테이션, 데모 또는 사람이 말하는 다른 모든 시나리오에 실시간 전사를 사용할 수 있습니다.

실시간 전사가 작동하려면 마이크에서 들어오는 오디오 또는 오디오 파일과 같은 다른 오디오 입력 소스를 애플리케이션에서 청취할 수 있어야 합니다. 애플리케이션 코드는 오디오를 서비스로 스트리밍하여 전사된 텍스트를 반환합니다.

전사 일괄 처리

모든 Speech to Text 시나리오가 실시간인 것은 아닙니다. 파일 공유, 원격 서버 또는 Azure 스토리지에 오디오 녹음이 저장되어 있을 수 있습니다. SAS(공유 액세스 서명) URI가 있는 오디오 파일을 가리키고 비동기적으로 전사 결과를 받을 수 있습니다.

일괄 작업이 '최선의 노력 기준'으로 예약되므로 전사 일괄 처리는 비동기 방식으로 실행해야 합니다. 일반적으로 작업은 요청 후 몇 분 이내에 실행을 시작하지만 작업이 실행 중 상태로 변경되는 시점에 대한 추정은 없습니다.

Text to Speech API

Text to Speech API를 사용하면 텍스트 입력을 가청 음성으로 변환할 수 있으며, 이를 컴퓨터 스피커를 통해 직접 재생하거나 오디오 파일에 쓸 수 있습니다.

음성 합성 목소리

Text to Speech API를 사용하는 경우 텍스트를 발음하는 데 사용할 음성을 지정할 수 있습니다. 이 기능을 통해 유연하게 음성 합성 솔루션을 개인화하고 개성을 부여할 수 있습니다.

이 서비스에는 '표준' 음성뿐만 아니라 '신경망'을 활용하여 억양과 관련된 음성 합성의 일반적인 한계를 극복하는 '신경' 음성을 포함하여 여러 언어 및 지역 발음을 지원하는 여러 미리 정의된 음성이 포함되어 있어 보다 자연스러운 음성이 생성됩니다. 사용자 지정 음성을 개발하고 Text to Speech API와 함께 사용할 수도 있습니다.

지원되는 언어

음성 텍스트 변환 및 Text to Speech API는 다양한 언어를 지원합니다. 아래 링크를 사용하여 지원되는 언어에 대한 세부 정보를 찾습니다.

- [Speech to Text 언어](#).
- [Text to Speech 언어](#).

다음 단원: 연습 - 음성 서비스 사용

계속 >

< 이전

단위 3/5 ▼

다음 >

✓ 100 XP



연습 - 음성 서비스 사용

20분

음성 인식 및 합성을 위해 음성 서비스 사용법을 배우는 가장 좋은 방법은 직접 탐색하는 것입니다.

시작하기 전에

이 연습을 완료하려면 다음이 필요합니다.

- Microsoft Azure 구독. 구독이 아직 없다면 <https://azure.microsoft.com/free>에서 평가판에 가입할 수 있습니다.
- MicrosoftDocs/ai-fundamentals GitHub 리포지토리에서 Jupyter Notebook을 실행할 수 있는 Python 환경. GitHub의 랩 파일을 자신의 Python 환경(있는 경우)에 복제하거나, 아래 지침에 따라 Azure 구독에서 Azure Machine Learning 작업 영역을 만들 수 있습니다.

ⓘ 참고

이 모듈은 Azure Machine Learning 작업 영역을 사용하는 여러 가지 모듈 중 하나입니다. **Azure AI 기본 사항** 인증의 준비 과정에서 이 모듈을 완료하는 경우, 작업 영역을 한 번 만들고 해당 인증 트랙의 모든 모듈에 다시 사용할 수 있습니다. 연습을 완료한 후에는 정리 지침에 따라 모듈 간 컴퓨팅 리소스를 중지해야 합니다.

Azure Machine Learning 작업 영역 만들기

Azure 구독에 Azure Machine Learning 작업 영역이 이미 있는 경우 새 브라우저 탭에서 [Azure Machine Learning Studio](#)로 이동하고 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다. 없는 경우 다음 단계에 따라 새 작업 영역을 만듭니다.

1. Azure 구독과 연결된 Microsoft 계정을 사용하여 [Azure Portal](#)에 로그인합니다.
2. + 리소스 만들기를 선택하고, *Machine Learning*을 검색한 뒤 다음 설정을 사용하여 새 *Machine Learning* 리소스를 만듭니다.
 - **작업 영역 이름:** 원하는 고유한 이름을 입력
 - **구독:** 자신의 Azure 구독
 - **리소스 그룹:** 고유한 이름의 새 리소스 그룹 만들기
 - **위치:** 사용 가능한 위치 선택

3. 작업 영역 리소스가 만들어질 때까지 기다립니다(몇 분이 걸릴 수 있음). 그런 다음 포털로 이동하고 작업 영역의 **개요** 페이지에서 Azure Machine Learning Studio를 실행(또는 새 브라우저 탭을 열고 <https://ml.azure.com>으로 이동)한 다음 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다.
4. Azure Machine Learning Studio에서 왼쪽 상단에 있는 **☰** 아이콘을 설정/해제하면 인터페이스에 다양한 페이지가 표시됩니다. 해당 페이지를 사용하여 작업 영역에서 리소스를 관리할 수 있습니다.

컴퓨팅 인스턴스 만들기

이 연습에서 사용된 Notebook을 실행하려면 Azure Machine Learning 작업 영역에 컴퓨팅 인스턴스가 필요합니다. 컴퓨팅 인스턴스가 이미 있는 경우 시작하고, 없는 경우 다음 지침에 따라 새로 만듭니다.

1. [Azure Machine Learning Studio](#)에서 **관리** 아래에 있는 **컴퓨팅** 페이지를 확인합니다.
2. **컴퓨팅 인스턴스** 탭에서 다음 설정을 사용하여 새 컴퓨팅 인스턴스를 만듭니다.
 - **지역**: 사용 가능한 지역 선택
 - **가상 머신 유형**: CPU
 - **가상 머신 크기**: Standard_DS11_v2
 - **컴퓨팅 이름**: 고유한 이름 입력
3. 컴퓨팅 인스턴스가 시작될 때까지 기다립니다(1분 정도 걸릴 수 있음).

연습 파일 다운로드

이 모듈(및 기타 관련 모듈)에 사용되는 파일은 GitHub 리포지토리에 게시되는데, 이를 자신의 Python 환경에 복제해야 합니다. 이전 모듈에서 **ai-fundamentals** 리포지토리를 아직 복제하지 않은 경우 다음 단계를 사용하여 Azure Machine Learning 작업 영역에 복제합니다.

1. [Azure Machine Learning Studio](#)에서 **Notebooks** 페이지(**작성자** 아래)를 확인합니다. 이 페이지에는 Notebooks 실행에 사용할 수 있는 Notebook 편집기가 포함되어 있습니다.
2. **내 파일**에서 **□** 단추를 사용하여 다음 설정에 따라 새 파일을 만듭니다.
 - **파일 위치**: `Users/your user name`
 - **파일 이름**: Get-Files
 - **파일 형식**: Notebook
 - **이미 있는 경우 덮어쓰기**: 선택됨
3. 새 Notebook이 만들어지면 이전에 만든 컴퓨팅 인스턴스가 **컴퓨팅** 상자에서 선택되었는지, 그리고 상태가 **실행 중**인지 확인합니다. 그런 다음 Notebook에서 만든 사각형 셀에 다음 코드를 붙여넣습니다.

```
!git clone https://github.com/MicrosoftDocs/ai-fundamentals
```

4. 셀 옆에 있는 ▶ 단추를 사용하여 포함된 코드를 실행합니다. 이렇게 하면 GitHub에서 연습 파일이 복제됩니다.
5. 코드의 실행이 끝나고 파일의 체크아웃이 완료되면 내 파일 아래에 있는 ⌂ 단추를 사용하여 폴더 보기를 새로 고치고, **ai-fundamentals**라는 이름의 폴더가 만들어졌는지 확인합니다. 이 폴더에는 연습에 사용된 Notebook 및 기타 파일이 포함되어 있습니다.
6. Get-Files.ipynb Notebook 탭을 닫습니다.

연습 완료

Python 환경을 설정하고 **ai-fundamentals** 리포지토리를 복제했으면 음성 인식 및 합성을 살펴볼 준비가 된 것입니다.

1. **ai-fundamentals** 폴더에서 **Speech.ipynb** Notebook을 엽니다. Azure Machine Learning Studio에서 Notebook 편집기를 사용하는 경우, << 단추로 파일 탐색기 창을 축소하여 Notebook 탭에 집중할 수 있는 공간을 좀 더 마련합니다.
2. Notebook의 정보를 읽고 포함된 코드 셀을 순서대로 실행합니다.

정리

Azure Machine Learning Studio에서 컴퓨팅 인스턴스를 사용한 경우 Azure 크레딧을 불필요하게 사용하지 않도록 이를 중지해야 합니다.

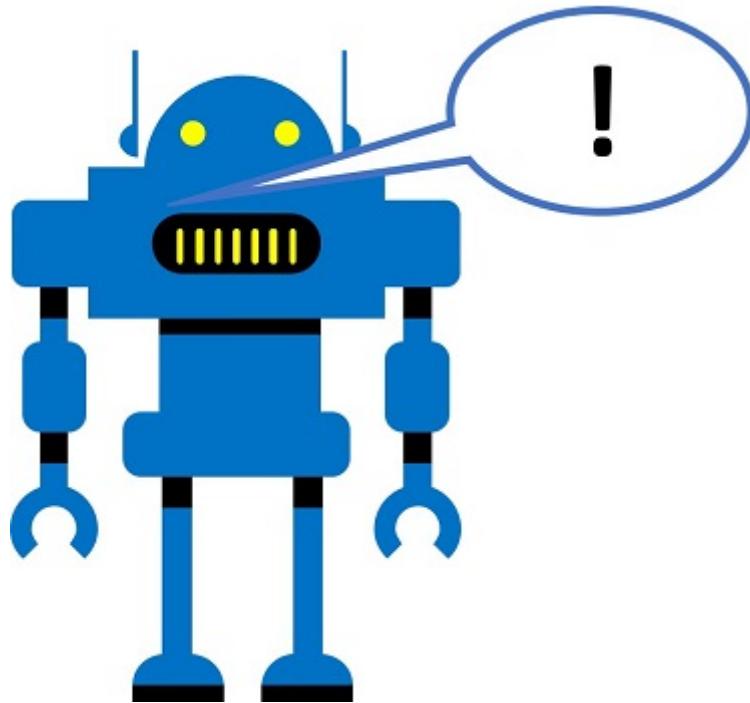
1. Azure Machine Learning Studio에서 관리 아래에 있는 컴퓨팅 페이지를 확인합니다.
2. 컴퓨팅 인스턴스 탭에서 컴퓨팅 인스턴스를 선택하고 중지 단추를 사용하여 중지합니다.

다음 단원: 지식 점검

계속 >

Speech 서비스

점점 더, 우리는 종종 음성 응답에 대한 기대를 가지고 인공지능(AI) 시스템과 대화함으로써 의사소통 할 수 있기를 기대한다.



음성 인식(*Speech Recognition*)(음성어를 해석하는 AI 시스템) 및 음성 합성(*Speedh Synthesis*)(음성 반응을 생성하는 AI 시스템)은 음성 지원 AI 솔루션의 핵심 구성 요소이다.

Cognitive Services 리소스 만들기

소리로된 음성을 해석하고 구두로 응답 할 수 있는 소프트웨어를 구축하기 위해서는 **Speech Cognitive** 서비스를 사용할 수 있는데 이 서비스는 간단한 방법으로 음성언어를 텍스트로 바꿀 수 있고 반대로 음성을 텍스트로 바꿀 수 있다.

이미 만들어진 Cognitive 서비스가 없다면 다음과 같은 순서로 Azure 구독에서 **Cognitive Services** 리소스를 생성할 수 있다.

1. 브라우저의 새로운 탭을 열고, Azure portal(<https://portal.azure.com>)을 (<https://portal.azure.com>)을 입력하고, Microsoft 계정으로 로그인한다..
2. + 리소스 만들기 버튼을 클릭하고, **Cognitive Services** 서비스를 찾은 다음, **Cognitive Services** 리소스를 다음과 같은 내용으로 생성한다.
 - **이름**: 유일한 이름을 입력한다(가능하면 영문과 숫자사용).
 - **구독**: Azure 구독선택.
 - **위치**: 가능한 위치(한국 중부 추천):
 - **가격책정계층**: 표준 S0
 - **리소스 그룹**: 원하는 유리한 이름(가능하면 영문과 숫자사용).
3. 배포가 완료될 때까지 기다린다. 그런 다음 Cognitive Services 리소스로 이동하여 *개요 페이지에서 링크를 클릭하여 서비스 키를 관리한다. 클라이언트 응용 프로그램에서 Cognitive Services 리소스에 연결하려면 엔드포인트와 키가 필요하다.

Cognitive Services 리소스에 있는 키와 엔드포인트 가져오기

Cognitive Services 리소스를 사용하기 위해서는, 클라이언트 응용프로그램에서는 엔드포인트와 인증 키가 필요하다.

1. Azure portal에서, Cognitive Services 리소스를 선택하고 **키 및 엔트포인트** 페이지를 선택한 다음 **키1** 을 복사하여 아래의 **YOUR_COG_KEY**.를 붙여 넣는다.
2. 리소스에 있는 **엔드포인트** 를 복사해서 아래의 **YOUR_COG_ENDPOINT**.에 붙여 넣는다.
3. 리소스에 있는 **위치** 를 복사해서 아래의 **YOUR_COG_REGION**.에 붙여 넣는다.
4. 아래 셀을 선택한 다음 셀 왼쪽에있는 **셀 실행(▷)** 버튼을 클릭하여 아래 코드를 실행한다.

In []:

```
cog_key = 'YOUR_COG_KEY'
cog_endpoint = 'YOUR_COG_ENDPOINT'
cog_region = 'YOUR_COG_REGION'

print('Ready to use cognitive services in {} using key {}'.format(cog_region, cog_key))
```

Cognitive 서비스에 이쓴 Speech 서비스를 이용하기 위해서는 Azure Cognitive Services Speech SDK를 설치해야 한다.K

In []:

```
! pip install azure.cognitiveservices.speech
```

음성인식(Speech recognition)

"전등을 켜라" 또는 "전등을 끄라"와 같은 음성 명령을 수신하는 홈 자동화 시스템을 구축하려고 한다고 가정하자. 응용 프로그램은 오디오 기반 입력(사용자의 음성 명령)을 가져와서 구문을 나누고 분석할 수 있는 텍스트로 변환하여 해석할 수 있어야 한다.

이제 여러분은 몇 가지 말을 글로 옮겨 쓰게 할 준비가 되었다. 입력은 마이크 또는 오디오 파일일 수 있다. 여기서는 오디오 파일을 사용한다.

아래 셀을 실행하여 Speech 서비스의 음성 텍스트 변환(Speech-to-text) 기능을 사용하여 오디오를 글로 변환한다.

In []:

```

import os
import IPython
from azure.cognitiveservices.speech import SpeechConfig, SpeechRecognizer, AudioConfig

# 오디오 파일로부터 음성 명령을 가져온다.
file_name = 'light-on.wav'
audio_file = os.path.join('data', 'speech', file_name)

# 음성 인식기를 설정한다
speech_config = SpeechConfig(cog_key, cog_region)
audio_config = AudioConfig(filename=audio_file) # 기본인 마이크 대신에 파일로 바꾼다
speech_recognizer = SpeechRecognizer(speech_config, audio_config)

# 일회성의 동기화되는 요청을 이용하여 음성명령을 글로 표현한다.
speech = speech_recognizer.recognize_once()

# 오디오를 실행하고 텍스트로 받아 쓴 명령을 보여준다.
IPython.display.display(IPython.display.Audio(audio_file, autoplay=True),
IPython.display.HTML(speech.text))

```

file_name 변수의 값을 *light-off.wav*로 바꾸고 셀을 다시 실행한다. 두개의 파일의 내용이 텍스트로 바르게 쓰여져야 한다.

음성합성(Speech synthesis)

이제 여러분은 스피치 서비스가 어떻게 말을 텍스트로 옮겨 쓸 수 있는지 보았을 것이다. 하지만 그 반대는 어떨까? 어떻게 텍스트를 음성으로 변환할 수 있을까?

在家 자동화 시스템이 조명을 켜는 명령을 해석했다고 가정해 보자. 적절한 응답은 구두로 명령을 확인하는 것일 수 있다(실제로 명령된 작업을 수행하는 것!).

In []:

```

import os
import IPython
from azure.cognitiveservices.speech import SpeechConfig, SpeechSynthesizer, AudioConfig

# 음성으로 보내질 텍스트를 가져온다.
response_text = 'Turning the light on.'

# 음성 학습 서비스를 설정한다
speech_config = SpeechConfig(cog_key, cog_region)
output_file = os.path.join('data', 'speech', 'response.wav')
audio_output = AudioConfig(filename=output_file) # Use a file instead of default (speakers)
speech_synthesizer = SpeechSynthesizer(speech_config, audio_output)

# 텍스트를 음성으로 바꾼다
result = speech_synthesizer.speak_text(response_text)

# 추출된 오디오 파일을 실행한다.
IPython.display.display(IPython.display.Audio(output_file, autoplay=True),
IPython.display.Image(data=os.path.join("data", "speech", response_text.lower() + '.jpg')))

```

response_text 변수값을 *Turning the light off.*(마지막에 마침표 포함)로 바꾸고 셀을 실행해서 결과를 들어보자.

심화학습

노트북에서 예제로 사용된 것은 매우 간단한 Speech Cognitive 서비스이다.

Speech 서비스내 음성 텍스트 변환 서비스에 대해서 좀 더 알아보고 싶다면 여기를 참고하라 [음성 텍스트 변환\(speech-to-text\)](https://docs.microsoft.com/azure/cognitive-services/speech-service/index-speech-to-text) (<https://docs.microsoft.com/azure/cognitive-services/speech-service/index-speech-to-text>).

그리고 Speech 서비스 내 텍스트 음성 변환 서비스에 대해서 알아보고 싶다면 다음을 참고하라 [텍스트 음성 변환 서비스\(text-to-speech\)](https://docs.microsoft.com/azure/cognitive-services/speech-service/index-text-to-speech) (<https://docs.microsoft.com/azure/cognitive-services/speech-service/index-text-to-speech>).

In []:



✓ 600 XP ➔

텍스트 및 음성 번역

29분 • 모듈 • 5 단위

★★★★★ 4.8 (542)

[평가하기](#)[초급](#) [AI 엔지니어](#) [데이터 과학자](#) [개발자](#) [솔루션 아키텍처](#) [학생](#)

Cognitive Services

AI 솔루션의 자동 번역 기능을 사용하면 언어 장벽을 허물어 보다 긴밀하게 협업할 수 있습니다.

학습 목표

이 모듈을 완료하면 Azure Cognitive Services를 사용하여 텍스트 및 음성 번역을 수행할 수 있습니다.

[책갈피](#) [컬렉션에 추가](#)

필수 조건

Azure Portal 탐색 능력

이 모듈은 이러한 학습 경로의 일부입니다.[자연어 처리 살펴보기](#)

소개

3분



Azure에서 번역 시작

3분



연습 - 텍스트 및 음성 번역

20분



지식 점검

2분



요약

1분



100 XP



소개

3분

조직과 개인이 다른 문화 및 지리적 위치에 있는 사람들과 협력해야 하는 경우 언어 장벽을 제거하는 것이 중요한 문제가 되었습니다.

한 가지 해결책은 2개 국어 또는 더 다양한 언어를 구사하는 사람을 찾는 것입니다. 그러나 이런 능력을 가진 사람들이 부족하다는 점과 가능한 언어 조합의 수에 대해 고려해보면, 이러한 접근 방식으로는 규모 확장이 어려울 수 있습니다. 이 문제를 해결하기 위해 기계 번역이라고도 하는 자동화된 번역이 점점 더 많이 사용되고 있습니다.

직역 및 의미론적 번역

초창기 기계 번역에는 직역이 적용되었습니다. 직역은 각 단어가 대상 언어의 해당 단어로 번역되는 것입니다. 이 방법에는 몇 가지 문제가 있습니다. 한 가지 예로, 대상 언어에 동등한 단어가 없을 수 있습니다. 또 다른 예시는 직역이 문장의 의미를 변경하거나 내용을 부정확하게 전달하는 경우입니다.

예를 들어 프랑스어 문구 "*éteindre la lumière*"가 영어로 "*turn off the light*"라고 번역될 수 있습니다. 그러나 프랑스어에서는 같은 의미로 "*fermer la lumiere*"라고 말할 수도 있습니다. 프랑스어 동사 *fermer*는 "*close*"를 의미하기 때문에 단어를 기준으로 하는 직역에서는 영어로 "*close the light*"가 됩니다. 일반적인 영어 사용자에게는 의미가 통하지 않습니다. 즉 번역 서비스 제공 시 의미론적 맥락을 고려해 "*turn off the light*"라는 영어 번역을 반환해야 합니다.

인공 지능 시스템은 단어뿐만 아니라 단어가 사용되는 의미론적 맥락을 이해할 수 있어야 합니다. 그러면 서비스 제공 시 입력 구문 또는 문장을 보다 정확하게 번역할 수 있습니다. 문법 규칙(공식/비공식) 및 구어체 모두가 고려 대상입니다.

텍스트 및 음성 번역

텍스트 번역은 한 언어에서 다른 언어로 문서를 번역하고, 외국 정부에서 온 이메일 내용을 번역하고, 인터넷에서 웹 페이지를 번역하는 기능을 제공하는 데 사용할 수 있습니다. 소셜 미디어 사이트의 게시물에 번역 옵션이 표시되거나, Bing 검색 엔진이 검색 결과에서 반환된 전체 웹 페이지 번역을 제공하는 것을 자주 보게 될 것입니다.

음성 번역은 음성 언어 간 번역에 사용되며, 곧바로 번역(음성 간 번역)되거나 중간 과정으로 텍스트 형식(음성 텍스트 변환 번역)을 거쳐 번역됩니다.

< 이전

단위 2/5 ▼

다음 >

100 XP



Azure에서 번역 시작

3분

Microsoft Azure는 번역을 지원하는 Cognitive Services를 제공합니다. 구체적으로는 다음과 같은 서비스를 이용할 수 있습니다.

- 텍스트 간 번역을 지원하는 **Translator Text** 서비스
- 음성 텍스트 변환 및 음성 간 번역을 가능하게 하는 **Speech** 서비스

Translator Text 및 Speech 관련 Azure 리소스

Translator Text 또는 Speech 서비스를 사용하려면 먼저 Azure 구독에 적절한 리소스를 프로비저닝해야 합니다.

이러한 서비스를 위한 전용 **Translator Text** 및 **Speech** 리소스 유형이 있으며 각 서비스의 액세스 및 청구를 개별적으로 관리하려는 경우에 사용할 수 있습니다.

또는 단일 Azure 리소스를 통해 두 서비스 모두에 대한 액세스를 제공하는 **Cognitive Services** 리소스를 만들어 청구를 통합하고 애플리케이션이 단일 엔드포인트 및 인증 키를 통해 두 서비스에 액세스 가능하도록 할 수 있습니다.

Translator Text 서비스를 사용하여 텍스트 번역

Translator Text 서비스는 애플리케이션, 웹 사이트, 도구 및 솔루션에 쉽게 통합할 수 있습니다. 이 서비스는 NMT(신경 기계 번역) 모델을 사용하여 번역하며, 텍스트의 의미론적 맥락을 분석하여 결과적으로 보다 정확하고 완전한 번역을 렌더링합니다.

Translator Text 서비스 지원 언어

Text Translator 서비스는 [60개 이상의 언어](#) 간에 텍스트 간 번역을 지원합니다. 이 서비스를 사용할 때는 영어의 경우 `_en*`, 프랑스어의 경우 `fr`, 중국어의 경우 `zh`와 같이 ISO 639-1 언어 코드를 사용하여 번역의 시작 언어(* `from` *)와 대상 언어(`to`)를 지정해야 합니다. 또는 미국 영어의 경우 `en-US`, 영국 영어의 경우 `en-GB`, 캐나다 프랑스어의 경우 `fr-CA`와 같이 보다 자세한 언어 코드인 3166-1 문화 코드를 적절히 사용해 문화권에 따른 언어 변형을 지정할 수도 있습니다.

Translator Text 서비스를 사용할 때는 하나의 시작 언어(* `from` *)와 여러 개의 대상 언어(`to`)를 지정하여 원본 문서를 동시에 여러 언어로 번역할 수 있습니다.

선택적 구성

Translator Text API에서 제공하는 다음과 같은 몇 가지 선택적 구성을 통해 반환되는 결과를 미세 조정할 수 있습니다.

- **욕설 필터링***. 구성을 하지 않으면 입력 텍스트에서 욕설을 필터링하지 않고 번역합니다. 욕설의 정도는 일반적으로 문화권에 따라 다르지만 번역된 텍스트를 욕설로 표시하거나 결과에서 생략하는 방식으로 욕설 번역을 제어할 수 있습니다.
- **선택적 번역**. 콘텐츠를 태그 지정하여 번역되지 않도록 할 수 있습니다. 예를 들어 지역화 하면 의미가 맞지 않게 되는 코드, 브랜드 이름 또는 단어/구에 태그를 지정할 수 있습니다.

Speech 서비스를 통한 음성 번역

Speech 서비스에는 다음과 같은 API(애플리케이션 프로그래밍 인터페이스)가 포함됩니다.

- **Speech-to-text** - 오디오 원본에서 텍스트 형식으로 음성을 전사하는 데 사용됩니다.
- **Text-to-speech** - 텍스트 소스에서 음성 오디오를 생성하는 데 사용됩니다.
- **Speech Translation** - 한 언어의 음성을 다른 언어의 텍스트 또는 음성으로 변환하는 데 사용됩니다.

Speech Translation API를 사용하여 마이크 또는 오디오 파일과 같은 스트리밍 원본의 음성 오디오를 번역하고, 번역본을 텍스트 또는 오디오 스트림으로 반환할 수 있습니다. 이렇게 하면 음성에 대한 실시간 자막 또는 음성 대화의 동시 양방향 번역과 같은 시나리오를 구현할 수 있습니다.

Speech 서비스 지원 언어

Translator Text 서비스와 마찬가지로, 하나의 소스 언어와 여러 개의 번역 대상 언어를 지정할 수 있습니다. 음성을 [60개 이상의 언어](#)로 번역할 수 있습니다.

소스 언어는 스페인어(라틴 아메리카)용 *es-US* 와 같이 확장 언어 및 문화 코드 형식을 사용하여 지정해야 합니다. 그러면 지역화된 발음과 언어 관용구를 사용할 수 있어 소스를 제대로 이해하는 데 도움이 됩니다.

대상 언어는 영어의 경우 *en*, 독일어의 경우 *de* 와 같이 두 글자로 구성된 언어 코드를 사용해야 합니다.

다음 단원: 연습 - 텍스트 및 음성 번역

계속 >

< 이전

단위 3/5 ▼

다음 >

✓ 100 XP



연습 - 텍스트 및 음성 번역

20분

번역에 Translator Text 및 Speech 서비스를 사용하는 방법을 배우는 가장 좋은 방법은 직접 해보는 것입니다.

시작하기 전 확인 사항

이 연습을 완료하려면 다음이 필요합니다.

- Microsoft Azure 구독. 구독이 아직 없다면 <https://azure.microsoft.com/free>에서 평가판에 가입할 수 있습니다.
- MicrosoftDocs/ai-fundamentals GitHub 리포지토리에서 Jupyter Notebook을 실행할 수 있는 Python 환경. GitHub의 랩 파일을 자신의 Python 환경(있는 경우)에 복제하거나, 아래 지침에 따라 Azure 구독에서 Azure Machine Learning 작업 영역을 만들 수 있습니다.

ⓘ 참고

이 모듈은 Azure Machine Learning 작업 영역을 사용하는 여러 가지 모듈 중 하나입니다.

Azure AI 기본 사항 인증의 준비 과정에서 이 모듈을 완료하는 경우, 작업 영역을 한 번 만들고 해당 인증 트랙의 모든 모듈에 다시 사용할 수 있습니다. 연습을 완료한 후에는 정리 지침에 따라 모듈 간 컴퓨팅 리소스를 중지해야 합니다.

Azure Machine Learning 작업 영역 만들기

Azure 구독에 Azure Machine Learning 작업 영역이 이미 있는 경우 새 브라우저 탭에서 [Azure Machine Learning Studio](#)로 이동하고 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다. 없는 경우 다음 단계에 따라 새 작업 영역을 만듭니다.

1. Azure 구독과 연결된 Microsoft 계정을 사용하여 [Azure Portal](#)에 로그인합니다.
2. + 리소스 만들기를 선택하고, *Machine Learning*을 검색한 뒤 다음 설정을 사용하여 새 *Machine Learning* 리소스를 만듭니다.
 - **작업 영역 이름:** 원하는 고유한 이름을 입력
 - **구독:** 자신의 Azure 구독
 - **리소스 그룹:** 고유한 이름의 새 리소스 그룹 만들기
 - **위치:** 사용 가능한 위치 선택

3. 작업 영역 리소스가 만들어질 때까지 기다립니다(몇 분이 걸릴 수 있음). 그런 다음 포털로 이동하고 작업 영역의 **개요** 페이지에서 Azure Machine Learning Studio를 실행(또는 새 브라우저 탭을 열고 <https://ml.azure.com>으로 이동)한 다음 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다.
4. Azure Machine Learning Studio에서 왼쪽 상단에 있는 **☰** 아이콘을 설정/해제하면 인터페이스에 다양한 페이지가 표시됩니다. 해당 페이지를 사용하여 작업 영역에서 리소스를 관리할 수 있습니다.

컴퓨팅 인스턴스 만들기

이 연습에서 사용된 Notebook을 실행하려면 Azure Machine Learning 작업 영역에 컴퓨팅 인스턴스가 필요합니다. 컴퓨팅 인스턴스가 이미 있는 경우 시작하고, 없는 경우 다음 지침에 따라 새로 만듭니다.

1. [Azure Machine Learning Studio](#)에서 **관리** 아래에 있는 **컴퓨팅** 페이지를 확인합니다.
2. **컴퓨팅 인스턴스** 탭에서 다음 설정을 사용하여 새 컴퓨팅 인스턴스를 만듭니다.
 - **지역**: 사용 가능한 지역 선택
 - **가상 머신 유형**: CPU
 - **가상 머신 크기**: Standard_DS11_v2
 - **컴퓨팅 이름**: 고유한 이름 입력
3. 컴퓨팅 인스턴스가 시작될 때까지 기다립니다(1분 정도 걸릴 수 있음).

연습 파일 다운로드

이 모듈(및 기타 관련 모듈)에 사용되는 파일은 GitHub 리포지토리에 게시되는데, 이를 자신의 Python 환경에 복제해야 합니다. 이전 모듈에서 **ai-fundamentals** 리포지토리를 아직 복제하지 않은 경우 다음 단계를 사용하여 Azure Machine Learning 작업 영역에 복제합니다.

1. [Azure Machine Learning Studio](#)에서 **Notebooks** 페이지(**작성자** 아래)를 확인합니다. 이 페이지에는 Notebooks 실행에 사용할 수 있는 Notebook 편집기가 포함되어 있습니다.
2. **내 파일**에서 **□** 단추를 사용하여 다음 설정에 따라 새 파일을 만듭니다.
 - **파일 위치**: `Users/your user name`
 - **파일 이름**: Get-Files
 - **파일 형식**: Notebook
 - **이미 있는 경우 덮어쓰기**: 선택됨
3. 새 Notebook이 만들어지면 이전에 만든 컴퓨팅 인스턴스가 **컴퓨팅** 상자에서 선택되었는지, 그리고 상태가 **실행 중**인지 확인합니다. 그런 다음 Notebook에서 만든 사각형 셀에 다음 코드를 붙여넣습니다.

```
!git clone https://github.com/MicrosoftDocs/ai-fundamentals
```

4. 셀 옆에 있는 ▶ 단추를 사용하여 포함된 코드를 실행합니다. 이렇게 하면 GitHub에서 연습 파일이 복제됩니다.
5. 코드의 실행이 끝나고 파일의 체크아웃이 완료되면 내 파일 아래에 있는 ⌂ 단추를 사용하여 폴더 보기를 새로 고치고, **ai-fundamentals**라는 이름의 폴더가 만들어졌는지 확인합니다. 이 폴더에는 연습에 사용된 Notebook 및 기타 파일이 포함되어 있습니다.
6. Get-Files.ipynb Notebook 탭을 닫습니다.

연습 완료

Python 환경을 설정하고 **ai-fundamentals** 리포지토리를 복제했으면 텍스트 음성 변환을 살펴볼 준비가 된 것입니다.

1. **ai-fundamentals** 폴더에서 **Translation.ipynb** Notebook을 엽니다. Azure Machine Learning Studio에서 Notebook 편집기를 사용하는 경우, <> 단추로 파일 탐색기 창을 축소하여 Notebook 탭에 집중할 수 있는 공간을 좀 더 마련합니다.
2. Notebook의 정보를 읽고 포함된 코드 셀을 순서대로 실행합니다.

정리

Azure Machine Learning Studio에서 컴퓨팅 인스턴스를 사용한 경우 Azure 크레딧을 불필요하게 사용하지 않도록 이를 중지해야 합니다.

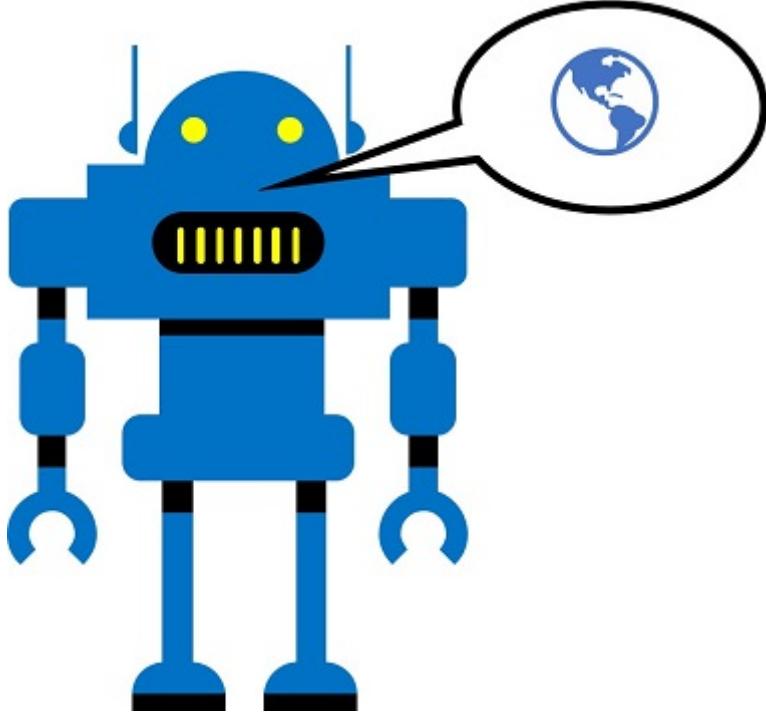
1. Azure Machine Learning Studio에서 관리 아래에 있는 컴퓨팅 페이지를 확인합니다.
2. 컴퓨팅 인스턴스 탭에서 컴퓨팅 인스턴스를 선택하고 중지 단추를 사용하여 중지합니다.

다음 단원: 지식 점검

계속 >

번역(Translation)

인류 문명이 발전할 수 있었던 원동력 중 하나는 서로 의사소통하는 능력이다. 대부분의 인간의 노력에서 의사소통이 핵심이다.



인공지능(AI)은 언어 간 텍스트 또는 음성을 번역하여 의사소통을 단순화하는 데 도움을 줄 수 있으며 국가 및 문화 간 의사소통의 장벽을 제거하는 데 도움이 될 수 있다..

Cognitive Services resource

Azure에서 여러 언어끼리 번역을 하기 위해서 Cognitive 서비스를 사용할 수 있다.

이미 만들어진 Cognitive 서비스가 없다면 다음과 같은 순서로 Azure 구독에서 **Cognitive Services** 리소스를 생성할 수 있다.

1. 브라우저의 새로운 탭을 열고, Azure portal(<https://portal.azure.com>)을 (<https://portal.azure.com>)을 입력하고, Microsoft 계정으로 로그인한다..
2. + 리소스 만들기 버튼을 클릭하고, Cognitive Services 서비스를 찾은 다음, **Cognitive Services** 리소스를 다음과 같은 내용으로 생성한다.
 - **이름**: 유일한 이름을 입력한다(가능하면 영문과 숫자사용).
 - **구독**: Azure 구독선택.
 - **위치**: 가능한 위치(한국 중부 추천):
 - **가격책정계층**: 표준 S0
 - **리소스 그룹**: 원하는 유리한 이름(가능하면 영문과 숫자사용).
3. 배포가 완료될 때까지 기다린다. 그런 다음 Cognitive Services 리소스로 이동하여 *개요 페이지에서 링크를 클릭하여 서비스 키를 관리한다. 클라이언트 응용 프로그램에서 Cognitive Services 리소스에 연결하려면 앤드포인트와 키가 필요하다.

Cognitive Services 리소스에 있는 키와 앤드포인트 가져오기

Cognitive Services 리소스를 사용하기 위해서는, 클라이언트 응용프로그램에서는 앤드포인트와 인증 키가 필요하다.

1. Azure portal에서 Cognitive Services 리소스를 선택하고 **키 및 엔트포인트** 페이지를 선택한 다음 **키1** 을 복사하여 아래의 **YOUR_COG_KEY**.를 붙여 넣는다.
2. 리소스에 있는 **엔드포인트** 를 복사해서 아래의 **YOUR_COG_ENDPOINT**.에 붙여 넣는다.
3. 리소스에 있는 **위치** 를 복사해서 아래의 **YOUR_COG_REGION**.에 붙여 넣는다.
4. 셀을 선택한 다음 셀 왼쪽에있는 **셀 실행(▷)** 버튼을 클릭하여 아래 코드를 실행한다.

In []:

```
cog_key = 'YOUR_COG_KEY'  
cog_endpoint = 'YOUR_COG_ENDPOINT'  
cog_region = 'YOUR_COG_REGION'  
  
print('Ready to use cognitive services in {} using key {}'.format(cog_region, cog_key))
```

텍스트 번역하기

Translator Text 서비스는 이름에서 알 수 있듯이 한 언어에서 다른 언어로 텍스트를 변환할 수 있도록 지원한다.

이 서비스에는 Python SDK가 없지만 REST 인터페이스를 사용하여 HTTP를 통해 엔드포인트에 요청을 제출할 수 있으며, **requests** 라이브러리를 사용하면 Python에서 비교적 쉽게 수행할 수 있다. 번역할 텍스트와 그 결과 번역된 텍스트에 대한 정보는 JSON 형식으로 교환한다.

다음 셀을 실행하여 이 작업을 수행하는 함수를 만든 다음 영어에서 프랑스어로 간단한 번역에 함수를 테스트해보자.

In []:

```
# Text Translation로 REST 요청을 보내는데 사용할 함수를 생성한다.
def translate_text(cog_region, cog_key, text, to_lang='fr', from_lang='en'):
    import requests, uuid, json

    # Text Translator 서비스 REST 요청을 위한 URL을 만든다.
    path = 'https://api.cognitive.microsofttranslator.com/translate?api-version=3.0'
    params = '&from={}&to={}'.format(from_lang, to_lang)
    constructed_url = path + params

    # Cognitive 리소스에 있는 키와 지역으로 요청 헤더를 구성한다.
    headers = {
        'Ocp-Apim-Subscription-Key': cog_key,
        'Ocp-Apim-Subscription-Region': cog_region,
        'Content-type': 'application/json',
        'X-ClientTraceId': str(uuid.uuid4())
    }

    # 번역할 텍스트를 Body에 추가한다.
    body = [
        'text': text
    ]

    # 번역본을 가져온다.
    request = requests.post(constructed_url, headers=headers, json=body)
    response = request.json()
    return response[0]["translations"][0]["text"]

# 함수를 테스트한다.
text_to_translate = "Hello"

translation = translate_text(cog_region, cog_key, text_to_translate, to_lang='fr', from_lang='en')
print('{} -> {}'.format(text_to_translate, translation))
```

이 서비스는 영어 텍스트 "Hello"를 프랑스어 "Bonjour"로 번역했어야 했다.

언어는 표준 언어 약어 시스템을 사용하여 지정되며, 영어의 경우 `en`, 프랑스어의 경우 `fr`를 사용한다(한국어는 `ko`이다). 특정 문화를 포함하는 약어를 사용할 수도 있다. 이 약어는 동일한 언어가 서로 다른 지리적 영역(대개 철자법)에서 사용될 때 유용하다. 예를 들어 `en-US`는 미국식 영어를, `en-GB`는 영국식 영어를 나타낸다.

다음 셀을 실행하여 영국 영어로 된 텍스트를 이탈리아어로 번역해보라.

In []:

```
text_to_translate = "Hello"

translation = translate_text(cog_region, cog_key, text_to_translate, to_lang='it-IT', from_lang='en-GB')
print('{} -> {}'.format(text_to_translate, translation))
```

이제 US 영어를 중국어로 번역하는 테스트를 해보자

In []:

```
text_to_translate = "Hello"

translation = translate_text(cog_region, cog_key, text_to_translate, to_lang='zh-CN', from_lang=
'en-US')
print('{} -> {}'.format(text_to_translate, translation))
```

음성 번역(Speech Translation)

Speech 서비스를 사용하여 구어를 번역할 수 있다.

Speech 서비스는 마이크나 오디오 파일에서 음성 입력을 변환하는 데 사용할 수 있는 Python SDK를 제공하므로 이를 설치해야한다..

In []:

```
! pip install azure.cognitiveservices.speech
```

이제 다음 셀을 실행하여 Speech SDK를 사용하여 소리 음성으로 번역하는 함수를 테스트 할 수 있다.

주의: 소리를 듣기 위해서 스피커가 필요하다.

In []:

```
# 하나의 언어에서 다른 언어로 텍스트를 번역하기 위한 함수를 만든다.
def translate_speech(cog_region, cog_key, audio_file=None, to_lang='fr-FR', from_lang='en-US'):
    from azure.cognitiveservices.speech import SpeechConfig, AudioConfig, ResultReason
    from azure.cognitiveservices.speech.translation import SpeechTranslationConfig, TranslationRecognizer

    # 음성 번역 서비스를 설정한다
    translation_config = SpeechTranslationConfig(subscription=cog_key, region=cog_region)
    translation_config.speech_recognition_language = from_lang
    translation_config.add_target_language(to_lang)

    # 음성 입력을 구성한다
    if audio_file is None:
        audio_config = AudioConfig() # 기본 입력을 사용(마이크)
    else:
        audio_config = AudioConfig(filename=audio_file) # 입력 파일 사용

    # 번역 인식도구를 생성하고 그것을 이용하여 음성 입력을 번역한다
    recognizer = TranslationRecognizer(translation_config, audio_config)
    result = recognizer.recognize_once()

    # 제대로 가져왔는가?
    translation = ''
    speech_text = ''
    if result.reason == ResultReason.TranslatedSpeech:
        speech_text = result.text
        translation = result.translations[to_lang]
    elif result.reason == ResultReason.RecognizedSpeech:
        speech_text = result.text
        translation = 'Unable to translate speech'
    else:
        translation = 'Unknown'
        speech_text = 'Unknown'

    # 번역본을 되돌려준다
    return speech_text, translation

# 함수를 테스트 한다
import os, IPython

file_name = 'english.wav'
file_path = os.path.join('data', 'translation', file_name)
speech, translated_speech = translate_speech(cog_region, cog_key, file_path, to_lang='es', from_lang='en-US')
result = '{} > {}'.format(speech, translated_speech)

# 요디오를 실행하고 번역된 텍스트를 나타낸다
IPython.display.display(IPython.display.Audio(file_path, autoplay=True),
                        IPython.display.HTML(result))
```

"to" 언어에는 2자로 된 언어 코드(예: `en`)를 사용하여 식별해야 하며, "from" 언어에는 문화 지표(예: `en-US`)가 포함되어야 한다(한국어의 경우 `ko-kr`임).

불어에서 영어로 번역해 보자

In []:

```
import os, IPython

file_name = 'french.wav'
file_path = os.path.join('data', 'translation', file_name)
speech, translated_speech = translate_speech(cog_region, cog_key, file_path, to_lang='en', from_
lang='fr-FR')
result = '{} → {}'.format(speech, translated_speech)

# 오디오를 실행하고 번역된 텍스트를 나타낸다.
IPython.display.display(IPython.display.Audio(file_path, autoplay=True),
IPython.display.HTML(result))
```

심화학습

번역에 대해서 좀더 살펴보려면 [텍스트 번역기\(Translator Text\)](https://docs.microsoft.com/azure/cognitive-services/translator/) (<https://docs.microsoft.com/azure/cognitive-services/translator/>)와 [Speech서비스를 활용한 음성번역 서비스](https://docs.microsoft.com/azure/cognitive-services/speech-service/index-speech-translation) (<https://docs.microsoft.com/azure/cognitive-services/speech-service/index-speech-translation>) 문서를 참조하기 바란다.



✓ 600 XP ➔

Language Understanding 을 사용하여 언어 모델 만들기

39분 • 모듈 • 5 단위

★★★★★ 4.8 (705)

[평가하기](#)

중급 AI 엔지니어 개발자 솔루션 아키텍처 학생 Azure Bot Service

Cognitive Services Language Understanding SDK

이 모듈에서는 Language Understanding 서비스에 대해 소개하고, 언어를 이해하는 애플리케이션을 만드는 방법을 보여 줍니다.

학습 목표

이 모듈에서는 다음을 수행합니다.

- Language Understanding이란 무엇인지 알아봅니다.
- 의도, 발화 등 주요 기능에 대해 알아봅니다.
- 자연어 기계 학습 모델을 빌드하고 게시합니다.

[챕터](#) 책갈피 [+ 컬렉션에 추가](#)

필수 조건

Azure Portal 탐색 능력

이 모듈은 이러한 학습 경로의 일부입니다.

[자연어 처리 살펴보기](#)

소개

3분



Language Understanding 시작하기

3분



연습 - Language Understanding 애플리케이션 만들기

30분



지식 점검

2분



요약

1분



✓ 100 XP



소개

3분

1950년, 영국의 수학자 앤런 튜링은 튜링 테스트로 알려진 모방 게임을 고안했으며, 대화가 충분히 자연스러우면 인간과 컴퓨터 중 누구와 대화하고 있는지 알 수 없게 된다는 가설을 세웠습니다. AI(인공 지능)가 점점 정교해지면서 애플리케이션 및 디지털 어시스턴트와 나누는 대화 방식의 상호 작용이 점점 보편화되었으며, 특정 시나리오에서는 AI 에이전트와 마치 인간처럼 상호 작용할 수 있게 되었습니다. 이런 솔루션의 일반적인 시나리오에는 고객 지원 애플리케이션, 예약 시스템, 홈 자동화 등이 있습니다.

모방 게임의 열망을 실현하려면, 컴퓨터는(텍스트 또는 오디오 형식으로) 언어 입력을 받아들일 수 있을 뿐 아니라 입력의 의미론적 의도를 해석할 수 있어야 합니다. 즉, 입력된 내용이 무슨 뜻인지 이해 할 수 있어야 합니다.

Microsoft Azure에서 언어 이해는 **Language Understanding Intelligent Service**를 통해 지원됩니다. 이 서비스는 일반적으로 **Language Understanding**으로 알려져 있습니다. Language Understanding을 사용하여 작업하려면 발화, 엔터티 및 의도라는 세 가지 핵심 개념을 이해해야 합니다.

발언

발화는 사용자가 소리 내어 말하는 무언가로, 애플리케이션이 해석해야 하는 대상입니다. 예를 들어 홈 자동화 시스템을 사용 중인 경우 사용자는 다음과 같은 발화를 사용할 수 있습니다.

“선풍기 켜 줘.”

“불 켜 줘.”

엔터티

엔터티는 발화가 참조하는 항목입니다. 예를 들어, 다음 발화에서 **선풍기** 와 **불** 이 엔터티에 해당합니다.

“_선풍기 켜 줘._”

“**불** 켜.”*

선풍기 및 **불** 엔터티를 **디바이스**라는 일반 엔터티의 특정한 인스턴스라고 생각할 수 있습니다.

의도

의도는 사용자의 발화에서 표현된 목적 또는 목표를 나타냅니다. 예를 들어 앞에서 살펴본 두 발화의 의도는 디바이스를 켜려는 것입니다. 따라서 Language Understanding 애플리케이션에서 해당 발화와 관련된 TurnOn 의도를 정의할 수 있습니다.

Language Understanding 애플리케이션은 의도와 엔터티로 구성된 모델을 정의합니다. 발화는 주어진 입력을 기반으로 하여 적용해야 하는 가장 적절한 의도와 엔터티를 식별하기 위해 모델을 학습시키는 데 사용됩니다. 앞에서 살펴본 험 도우미 애플리케이션에는 다음 예시와 같은 여러 가지 의도가 포함될 수 있습니다.

Intent	관련 발화	엔터티
Greeting	"Hello" "Hi" "있잖아"	
TurnOn	"선풍기 켜 줘" "불 켜 줘" "불 켜"	선풍기(디바이스) 불(디바이스) 불(디바이스)
TurnOff	"선풍기 꺼 줘" "불 꺼 줘" "불 꺼"	선풍기(디바이스) 불(디바이스) 불(디바이스)
CheckWeather	"오늘 날씨 어때?" "일기 예보 알려 줘" "파리 일기 예보는 어때?" "내일 시애틀의 날씨는 어때?"	오늘(날짜/시간) 오늘(날짜/시간) 파리(위치) 시애틀(위치), 내일(날짜/시간)
None	"삶의 의미는 무엇인가?" "이거 켜진 건가?"	

위 표에는 각 의도를 위해 사용되는 여러 가지 발화가 있습니다. 의도는 발화 작업을 그룹화하는 간결한 방법이어야 합니다. 여기서 특별 관심 분야는 '*None*' 의도입니다.*_ 입력한 발화 중 어떤 것과도 매핑되지 않는 발화를 처리하기 위해 항상 *None* 의도를 사용하는 것이 좋습니다. *None* 의도는 대체 옵션으로 간주되며, 일반적으로 사용자의 요청이 다른 어떤 의도와도 일치하지 않는 경우 사용자에게 일반적인 응답을 제공하는 데 사용됩니다.

💡 팁

Language Understanding 애플리케이션에서 *_None* * 의도는 생성되긴 하지만 의도적으로 비어 있습니다. *None* 의도는 필수 의도이며 삭제하거나 이름을 바꿀 수 없습니다. 이 의도를 도메인 외부에 있는 발화로 채웁니다.

Language Understanding 애플리케이션에서 샘플 발화로 엔터티 및 의도를 정의한 후에는 사용자 입력이 샘플 발화와 정확히 일치하지 않더라도 사용자 입력으로부터 의도 및 엔터티를 예측하도록 언어 모델을 학습시킬 수 있습니다. 그런 다음 클라이언트 애플리케이션에서 모델을 사용하여 예측을 가져오고 그것에 맞게 응답할 수 있습니다.

다음 단원: Language Understanding 시작하기

계속 >

< 이전

단위 2/5 ▼

다음 >

✓ 100 XP



Language Understanding 시작하기

3분

Language Understanding을 사용하여 언어 이해 애플리케이션을 만드는 것은 두 가지 주요 작업으로 구성됩니다. 먼저 언어 모델을 학습시키는 데 사용할 엔터티, 의도 및 발화를 정의해야 합니다. 이 단계를 모델을 '제작'한다고 합니다. 그런 다음 클라이언트 애플리케이션이 사용자 입력을 기반으로 하여 의도 및 엔터티 '예측'에 사용할 수 있도록 모델을 게시해야 합니다.

Language Understanding용 Azure 리소스

각각의 제작 및 예측 작업에 대해 Azure 구독의 리소스가 필요합니다. 다음과 같은 유형의 리소스를 사용할 수 있습니다.

- **Language Understanding** : 작성 또는 예측 리소스가 될 수 있는 Language Understanding 전용 리소스입니다.
- **Cognitive Services** : Language Understanding을 비롯해 다른 많은 인지 서비스를 포함하는 일반적인 인지 서비스 리소스입니다. 예측을 위해서는 이 종류의 리소스만 사용할 수 있습니다.

작성 리소스와 예측 리소스의 분리는 예측을 생성하기 위해 모델을 사용하는 클라이언트 애플리케이션과 별도로 언어 모델 학습의 리소스 사용률을 추적하려는 경우에 유용합니다.

Language Understanding 리소스를 만들도록 선택하면 작성, 예측 또는 둘 다를 선택하라는 메시지가 표시됩니다. '둘 다'를 선택하면 작성을 위한 리소스와 예측을 위한 리소스가 각각 하나씩 생성됩니다.*

또는 작성을 위한 전용 Language Understanding 리소스를 사용하고, 예측을 위해서는 모델을 일반 Cognitive Services 리소스에 배포할 수도 있습니다. 클라이언트 애플리케이션이 Language Understanding 이외에 다른 인지 서비스를 사용하는 경우, 이 방법을 사용하면 단일 엔드포인트와 키를 통해 Language Understanding 예측 서비스를 포함하여 사용 중인 모든 인지 서비스에 대한 액세스를 관리할 수 있습니다.

작성

작성 리소스를 만들었으면 예측 모델을 학습시키는 데 사용할 수 있도록 애플리케이션이 예측 할 엔터티 및 의도와 각 의도에 대한 발화를 정의하여 Language Understanding 애플리케이션을 작성하고 학습시키는 데 이 리소스를 사용할 수 있습니다.

Language Understanding은 일반적인 시나리오에 대해 미리 정의된 의도 및 엔터티를 포함하는 사전 빌드된 '도메인'의 포괄적인 컬렉션을 제공하며, 이 도메인은 모델의 시작점으로 사용할 수 있습니다._* 사용자 고유의 엔터티 및 의도를 만들 수도 있습니다.

엔터티 및 의도를 만들 때는 어떤 순서로든 이를 수행할 수 있습니다. 의도를 만든 다음 의도에 대해 정의한 샘플 발화에서 단어를 선택하여 엔터티를 만들 수도 있고, 엔터티를 미리 만든 다음 의도를 만드는 과정에서 발화에 포함된 단어에 의도를 매핑할 수도 있습니다.

모델의 요소를 정의하기 위해 코드를 작성할 수 있지만, 대부분의 경우 Language Understanding 애플리케이션을 만들고 관리하기 위한 웹 기반 인터페이스인 Language Understanding 포털을 사용하여 모델을 작성하는 것이 가장 쉽습니다.

[팁] 모범 사례는 Language Understanding 포털을 작성에 사용하고 SDK를 런타임 예측에 사용하는 것입니다.

의도 만들기

사용자가 애플리케이션을 사용하여 어떤 작업을 수행할지에 따라 의도를 정의합니다. 각 의도에 대해 사용자가 의도를 어떻게 표현하면 좋은지에 대한 예제를 제공하는 다양한 발화를 포함해야 합니다.

의도를 여러 엔터티에 적용할 수 있는 경우, 각 잠재 엔터티에 대해 샘플 발화를 포함해야 하며, 각 엔터티가 발화에서 식별되어야 확인합니다.

엔터티 만들기

엔터티에는 네 가지 유형이 있습니다.

- **Machine-Learned** : 제공된 샘플 발화의 컨텍스트로부터 학습하는 동안 모델에 의해 학습된 엔터티입니다.
- **List** : 목록 및 하위 목록의 계층 구조로 정의된 엔터티입니다. 예를 들어 **디바이스** 목록에는 **불**과 **선풍기**의 하위 목록이 포함될 수 있습니다. 각 목록 항목에 대해 **불**의 경우 **조명**과 같은 동의어를 지정할 수 있습니다.
- **RegEx** : 패턴을 설명하는 '정규식'으로 정의된 엔터티입니다. 예를 들어 *_555-123-4567 형식의 전화번호는 [0-9]{3}-[0-9]{3}-[0-9]{4}와 같은 패턴으로 정의할 수 있습니다.
- **_Pattern.any**: 샘플 발화에서 추출하기 어려울 수 있는 복잡한 엔터티를 정의하기 위해 '패턴'과 함께 사용되는 엔터티입니다.

모델 학습

모델의 의도와 엔터티를 정의하고 적절한 샘플 발화 세트를 포함했다면 이제 모델을 학습시킬 차례입니다. 학습이란 샘플 발화를 사용하여 사용자가 말할 수 있는 의도 및 엔터티의 자연어

표현과 일치하도록 모델을 가르치는 프로세스입니다.

모델을 학습시킨 후에는 텍스트를 제출하고 예측된 의도를 검토하여 모델을 테스트할 수 있습니다. 학습과 테스트는 반복적인 프로세스입니다. 모델을 학습시킨 후에는 샘플 발화로 모델을 테스트하여 의도와 엔터티가 제대로 인식되는지 확인합니다. 제대로 인식되지 않으면 업데이트하고 다시 학습시킨 후에 다시 테스트합니다.

예측

학습 및 테스트 결과에 만족하면 Language Understanding 애플리케이션이 사용될 수 있도록 예측 리소스에 게시할 수 있습니다.

클라이언트 애플리케이션은 예측 리소스의 엔드포인트에 연결하고 적절한 인증 키를 지정하고 사용자 입력을 제출하여 예측된 의도 및 엔터티를 얻음으로써 모델을 사용할 수 있습니다. 예측은 클라이언트 애플리케이션으로 반환되며, 클라이언트 애플리케이션은 예측된 의도에 따라 적절한 조치를 할 수 있습니다.

다음 단원: 연습 - Language Understanding 애플리케이션 만들기

[계속 >](#)

< 이전

단위 3/5 ▼

다음 >

✓ 100 XP



연습 - Language Understanding 애플리케이션 만들기

30분

Language Understanding을 사용하는 방법을 배우는 가장 좋은 방법은 직접 살펴보는 것입니다.

시작하기 전에

이 연습을 완료하려면 다음이 필요합니다.

- Microsoft Azure 구독. 구독이 아직 없다면 <https://azure.microsoft.com/free>에서 평가판에 가입할 수 있습니다.
- MicrosoftDocs/ai-fundamentals GitHub 리포지토리에서 Jupyter Notebook을 실행할 수 있는 Python 환경. GitHub의 랩 파일을 자신의 Python 환경(있는 경우)에 복제하거나, 아래 지침에 따라 Azure 구독에서 Azure Machine Learning 작업 영역을 만들 수 있습니다.

① 참고

이 모듈은 Azure Machine Learning 작업 영역을 사용하는 여러 가지 모듈 중 하나입니다.

Azure AI 기본 사항 인증의 준비 과정에서 이 모듈을 완료하는 경우, 작업 영역을 한 번 만들고 해당 인증 트랙의 모든 모듈에 다시 사용할 수 있습니다. 연습을 완료한 후에는 정리 지침에 따라 모듈 간 컴퓨팅 리소스를 중지해야 합니다.

Azure Machine Learning 작업 영역 만들기

Azure 구독에 Azure Machine Learning 작업 영역이 이미 있는 경우 새 브라우저 탭에서 [Azure Machine Learning Studio](#)로 이동하고 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다. 없는 경우 다음 단계에 따라 새 작업 영역을 만듭니다.

- Azure 구독과 연결된 Microsoft 계정을 사용하여 [Azure Portal](#)에 로그인합니다.
- + 리소스 만들기를 선택하고, *Machine Learning*을 검색한 뒤 다음 설정을 사용하여 새 Machine Learning 리소스를 만듭니다.
 - 작업 영역 이름:** 원하는 고유한 이름을 입력
 - 구독:** 자신의 Azure 구독
 - 리소스 그룹:** 고유한 이름의 새 리소스 그룹 만들기

- **위치:** 사용 가능한 위치 선택

3. 작업 영역 리소스가 만들어질 때까지 기다립니다(몇 분이 걸릴 수 있음). 그런 다음 포털로 이동하고 작업 영역의 **개요** 페이지에서 Azure Machine Learning Studio를 실행(또는 새 브라우저 탭을 열고 <https://ml.azure.com> 으로 이동)한 다음 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다.
4. Azure Machine Learning Studio에서 왼쪽 상단에 있는 ⓢ 아이콘을 설정/해제하면 인터페이스에 다양한 페이지가 표시됩니다. 해당 페이지를 사용하여 작업 영역에서 리소스를 관리할 수 있습니다.

컴퓨팅 인스턴스 만들기

이 연습에서 사용된 Notebook을 실행하려면 Azure Machine Learning 작업 영역에 컴퓨팅 인스턴스가 필요합니다. 컴퓨팅 인스턴스가 이미 있는 경우 시작하고, 없는 경우 다음 지침에 따라 새로 만듭니다.

1. [Azure Machine Learning Studio](#) 에서 **관리** 아래에 있는 **컴퓨팅** 페이지를 확인합니다.
2. **컴퓨팅 인스턴스** 탭에서 다음 설정을 사용하여 새 컴퓨팅 인스턴스를 만듭니다.
 - **지역:** 사용 가능한 지역 선택
 - **가상 머신 유형:** CPU
 - **가상 머신 크기:** Standard_DS11_v2
 - **컴퓨팅 이름:** 고유한 이름 입력
3. 컴퓨팅 인스턴스가 시작될 때까지 기다립니다(1분 정도 걸릴 수 있음).

연습 파일 다운로드

이 모듈(및 기타 관련 모듈)에 사용되는 파일은 GitHub 리포지토리에 게시되는데, 이를 자신의 Python 환경에 복제해야 합니다. 이전 모듈에서 **ai-fundamentals** 리포지토리를 아직 복제하지 않은 경우 다음 단계를 사용하여 Azure Machine Learning 작업 영역에 복제합니다.

1. [Azure Machine Learning Studio](#) 에서 **Notebooks** 페이지(**작성자** 아래)를 확인합니다. 이 페이지에는 Notebook 실행에 사용할 수 있는 Notebook 편집기가 포함되어 있습니다.
2. **내 파일**에서 □ 단추를 사용하여 다음 설정에 따라 새 파일을 만듭니다.
 - **파일 위치:** Users/your user name
 - **파일 이름:** Get-Files
 - **파일 형식:** Notebook
 - **이미 있는 경우 덮어쓰기:** 선택됨
3. 새 Notebook이 만들어지면 이전에 만든 컴퓨팅 인스턴스가 **컴퓨팅** 상자에서 선택되었는지, 그리고 상태가 **실행 중**인지 확인합니다. 그런 다음 Notebook에서 만든 사각형 셀에

다음 코드를 붙여넣습니다.

복사

```
!git clone https://github.com/MicrosoftDocs/ai-fundamentals
```

4. 셀 옆에 있는 ▶ 단추를 사용하여 포함된 코드를 실행합니다. 이렇게 하면 GitHub에서 연습 파일이 복제됩니다.
5. 코드의 실행이 끝나고 파일의 체크아웃이 완료되면 내 파일 아래에 있는 ⌂ 단추를 사용하여 폴더 보기를 새로 고치고, **ai-fundamentals**라는 이름의 폴더가 만들어졌는지 확인합니다. 이 폴더에는 연습에 사용된 Notebook 및 기타 파일이 포함되어 있습니다.
6. Get-Files.ipynb Notebook 탭을 닫습니다.

연습 완료

Python 환경을 설정하고 **ai-fundamentals** 리포지토리를 복제했으면 Language Understanding을 살펴볼 준비가 된 것입니다.

1. **ai-fundamentals** 폴더에서 **Language Understanding.ipynb** Notebook을 엽니다. Azure Machine Learning Studio에서 Notebook 편집기를 사용하는 경우, <> 단추로 파일 탐색기 창을 축소하여 Notebook 탭에 집중할 수 있는 공간을 좀 더 마련합니다.
2. Notebook의 정보를 읽고 포함된 코드 셀을 순서대로 실행합니다.

정리

Azure Machine Learning Studio에서 컴퓨팅 인스턴스를 사용한 경우 Azure 크레딧을 불필요하게 사용하지 않도록 이를 중지해야 합니다.

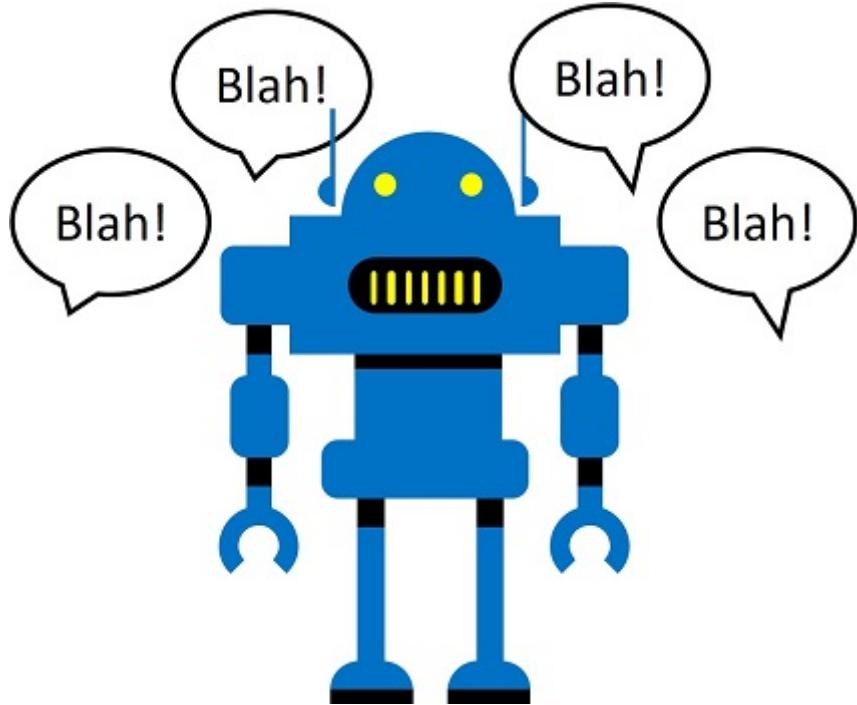
1. Azure Machine Learning Studio에서 관리 아래에 있는 컴퓨팅 페이지를 확인합니다.
2. 컴퓨팅 인스턴스 탭에서 컴퓨팅 인스턴스를 선택하고 중지 단추를 사용하여 중지합니다.

다음 단원: 지식 점검

계속 >

언어 이해(Language Understanding)

점점 더, 우리는 컴퓨터가 자연 언어로 말하거나 입력된 명령을 이해하기 위해 AI를 사용할 수 있기를 기대한다. 예를 들어, "전등을 켜라" 또는 "팬을 켜라"와 같은 음성 명령을 사용하여 가정에서 장치를 제어할 수 있도록 하는 홈 자동화 시스템을 구현하고 AI로 구동되는 장치가 명령을 이해하고 적절한 조치를 취하도록 할 수 있다.



작성 및 예측 리소스 만들기

Microsoft Cognitive 서비스에는 발언(*Utterance*)에 따라 엔터티에 적용되는 의도(*intent*)를 정의할 수 있는 Language Understanding 서비스가 포함되어 있습니다. **Language Understanding** 또는 **Cognitive 서비스** 리소스를 사용하여 Language Understanding 앱을 게시할 수 있지만 작성을 위해 별도의 **Language Understanding** 리소스를 생성해야 한다.

1. A. 브라우저의 새로운 탭을 열고, Azure portal(<https://portal.azure.com>)을 (<https://portal.azure.com>)을) 입력하고 , Microsoft 계정으로 로그인한다..
2. + 래소스 만들기를 클릭하고 *Language Understanding*을 찾는다..
3. 서비스 목록에서 **Language Understanding**을 클릭한다.
4. **Language Understanding** 을 선택하고 만들기를 누른다.
5. 만들기 페이지에서 다음 내용을 입력하고 만들기를 클릭한다.
 - **만들기 옵션:** 둘 다
 - **이름:** 유일한 서비스 이름
 - **구독:** Azure 구독서비스를 선택
 - **리소스 그룹:** 이미 존재하는 리소스 그룹을 선택하거나 새로운 것을 만듬
 - **작성 위치:** 사용 가능한 위치를 선택
 - **작성 가격 책정 계층:** F0
 - **예측 위치:** 작성 위치와 같은 위치 선택
 - **예측 가격 책정 계층:** F0 6. 리소스가 만들어지까지 기다리면 두 개의 Language Understanding 리소스 가 만들어지게 되는데 하나는 제작용이고 다른 하나는 예측용이다. 만들어진 리소스 그룹을 클릭하여 각 리소스를 확인할 수 있다.

Language Understanding 앱 만들기

Language Understanding를 통해 자연어 Language Understanding을 구현하려면 앱을 만든 다음 엔터티, 의도(intents) 및 발언(Utterance)을 추가하여 앱에서 이해할 명령들을 정의한다.

1. 브라우저의 새로운 탭에서 Language Understanding 포털 <https://www.luis.ai> (<https://www.luis.ai>) 을 열고 Azure 구독과 관련된 Microsoft 계정으로 로그인을 한다. 만일 계정이 Language Understading 포털에 처음으로 로그인을 하였다면 계정 세부 정보에 액세스가 가능할록 권한을 부여하도록 하는 작업이 필요하다. Azure 구독에서 만들었던 Language Understanding 작성 리소스를 선택함으로 *Welcome* 과정을 완료한다.
2. **My Apps** 페이지를 클릭하고 구독을 선택하고 Language Understanding 작성 리소스를 선택한다. 그리고 나서 다음과 같은 설정으로 Conversation을 위한 새로운 앱을 생성한다.
 - **Name:** Home Automation
 - **Culture:** English
 - **Description:** 간단한 홈 자동화시스템
 - **Prediction resource:** 앞에서 만든 *Language Understanding* 예측 리소스
3. 효과적인 Language Understanding 앱을 만드는데 필요한 팁 메뉴가 나타나면 닫는다.

엔터티(Entity) 만들기

*entity*는 언어 모델이 식별하여 수행할 수 있는 작업이다. 이 경우 Language Understanding 앱은 조명이나 팬과 같은 사무실의 다양한 기기를 제어하는 데 사용되므로, 앱에서 사용할 장치 유형 목록이 포함된 *devices* 엔티티를 만든다. 각 장치 유형에 대해 장치의 이름(예: *light*)과 이 장치 유형을 참조하는 데 사용할 수 있는 모든 동의어(예: *lamp*)를 식별하는 하위 목록을 생성한다.

1. 앱의 Language Understanding 페이지에서, 왼쪽에 있는 **Entities**를 클릭한다. 그리고 나서 **Create**를 클릭하고 새로운 엔터티 이름으로 **device**라 넣고, **List** 형식을 선택하고 **Create**를 클릭한다..
2. **List items** 페이지에서 **Normalized Values** 값 밑에 **light**라 넣고 ENTER를 누른다..
3. **light** 값이 추가된 뒤에 **Synonyms** 밑에 **lamp**라 입력하고 ENTER를 누른다..
4. 두 번째 리스트 항목으로 **fan**을 넣고 동의어로 **AC**을 입력한다.

의도(Intents) 만들기

*Intents*는 하나 이상의 엔티티에 대해 수행할 작업이다. 예를 들어 조명을 켜거나 팬을 끌 수 있다. 이 경우 장치를 켜 때와 장치를 끌 때의 두 가지 의도를 정의한다. 각 의도에 대해 Intents를 나타내는 언어 종류를 나타내는 표본 *Utterance*를 지정한다.

1. 페이지 왼편에 있는 **Intents**를 선택하고, **Create** 클릭하고 **switch_on**이라는 이름으로 intent를 추가한다. 그리고 나서 **Done**을 클릭한다..
2. **Examples**와 **Example user input**의 밑으로 이동하여 **turn the light on**이라 utterance를 입력하고 **Enter**를 눌러 이것을 리스트에 포함시킨다.
3. **turn the light on** utterance에서 "light" 단어를 클릭하고 **device** 엔터티의 **light** 값을 할당한다.
4. **switch_on** intent의 두 번째 utterance로 **turn the fan on**을 입력한다. 그리고 나서 "fan" 단어를 클릭하고 **device** 엔터티의 **fan** 값을 할당한다.
5. 페이지 왼편에 있는 **Intents**를 선택하고, **Create** 클릭하고 **switch_off**이라는 이름으로 intent를 추가한다
6. **switch_off** intent에 대한 페이지로 이동하여 **turn the light off** utterance를 입력하고 "light" 단어에 **device** 엔터티의 **light** 값을 할당한다.
7. **switch_off** intent의 두 번째 utterance로 **turn the fan off** 구를 입력한다. 그리고 나서 "fan" 단어에 **device** 엔터티의 **fan** 값을 할당한다.

언어 모델을 학습시키고 테스트하기

이제 엔터티들, intents와 utterance의 형식으로 제공한 데이터를 이용하여 앱의 언어모델을 학습시킨다.

1. 앱의 Language Understanding 페이지에서 **Train**을 클릭하여 언어 모델을 학습시킨다.
2. 모델이 하였다면 **Test**를 클릭하고 아래의 구들을 입력하여 예측된 intent를 Test페이지에서 확인한다.
 - *switch the light on*
 - *turn off the fan*
 - *turn the lamp off*
 - *switch on the AC*
3. 테스트 창을 닫는다.

모델을 게시하고 엔드포인트를 설정하기

학습시킨 모델을 클라이언트 응용프로그램에서 사용하기 위해서는 클라이언트 응용프로그램이 새로운 utterance이 보낼 수 있는 엔드포인트로 게시해야 한다. 이렇게 하면 intent와 엔터티들을 예측할 수 있다.

1. 앱에서 Language Understanding 페이지에 있는 **Publish**를 클릭한다. 그런 다음 **Production slot** 클릭하고 **Done**을 선택한다..
2. 모델이 게시된 후 앱의 Language Understanding 페이지의 위에 있는 **Manage**를 클릭한다. 그리고 나서 **Application Information** 탭에서, 앱의 **App ID**를 확인한다. 이것을 복사해서 아래의 셀에 있는 **YOUR_LU_APP_ID**에 붙여넣기를 한다..
3. **Azure Resources** 탭에서 예측 리소스의 **Primary key**와 **Endpoint URL**을 확인한다. 복사하여 아래 코드의 **YOUR_LU_KEY**와 **YOUR_LU_ENDPOINT**에 대신하여 붙여넣기를 한다..
4. 아래 셀의 **Run cell** (▷) 버튼(왼쪽에 있음)을 클릭하여 실행한다. 입력창이 나타나 *turn the light on*라고

In []:

```
from python_code import luis
import matplotlib.pyplot as plt
from PIL import Image
import os
%matplotlib inline

try:
    # API 구성 설정하기
    luis_app_id = 'YOUR_LU_APP_ID'
    luis_key = 'YOUR_LU_KEY'
    luis_endpoint = 'YOUR_LU_ENDPOINT'

    # 명령창 띄우기
    command = input('Please enter a command: \n')

    # 예측된 intent와 엔터티 가져오기(python_code.home_auto.py 코드에 있음)
    action = luis.get_intent(luis_app_id, luis_key, luis_endpoint, command)

    # 알맞은 이미지 보여주기
    img_name = action + '.jpg'
    img = Image.open(os.path.join("data", "luis", img_name))
    plt.axis('off')
    plt.imshow(img)
except Exception as ex:
    print(ex)
```

아래와 같은 구로 위를 다시 실행해 보자:

- *turn on the light*
- *put the lamp off*
- *switch the fan on*
- *switch the light on*
- *switch off the light*
- *turn off the fan*
- *switch the AC on*

알림: 만일 Language Understanding 앱이 intents와 엔터티를 가져오는 방법에 대해서 궁금하다면 **python_code** 폴더의 **luis.py** 파일을 참고하라.

음성 제어(Voice Control) 추가하기

지금까지 우리는 텍스트를 분석하는 방법을 살펴보았다. 하지만 점점 더 AI 시스템은 음성 인식을 통해 인간이 소프트웨어 서비스와 의사소통할 수 있게 한다. 이를 지원하기 위해 **Speech** Cognitive 서비스는 음성을 텍스트로 간단하게 기록할 수 있는 방법을 제공한다.

Cognitive 서비스 리소스 만들기

이미 만들어진 Cognitive 서비스가 없다면 다음과 같은 순서로 Azure 구독에서 **Cognitive Services** 리소스를 생성할 수 있다.

1. 브라우저의 새로운 탭을 열고, Azure portal(<https://portal.azure.com>)을 (<https://portal.azure.com>)을 입력하고, Microsoft 계정으로 로그인한다..
2. + 리소스 만들기 버튼을 클릭하고, **Cognitive Services** 서비스를 찾은 다음, **Cognitive Services** 리소스를 다음과 같은 내용으로 생성한다.
 - **이름:** 유일한 이름을 입력한다(가능하면 영문과 숫자 사용).
 - **구독:** Azure 구독선택.
 - **위치:** 가능한 위치(한국 중부 추천):
 - **가격책정계층:** 표준 S0
 - **리소스 그룹:** 원하는 유리한 이름(가능하면 영문과 숫자 사용).
3. 배포가 완료될 때까지 기다린다. 그런 다음 Cognitive Services 리소스로 이동하여 *개요 페이지에서 링크를 클릭하여 서비스 키를 관리한다. 클라이언트 응용 프로그램에서 Cognitive Services 리소스에 연결하려면 엔드포인트와 키가 필요하다.

Cognitive Services 리소스에 있는 키와 엔드포인트 가져오기

Cognitive Services 리소스를 사용하기 위해서는, 클라이언트 응용 프로그램에서는 엔드포인트와 인증 키가 필요하다.

1. Azure portal에서, Cognitive Services 리소스를 선택하고 **키 및 엔트포인트** 페이지를 선택한 다음 **키1**을 복사하여 아래의 **YOUR_COG_KEY**를 붙여 넣는다.
2. 리소스에 있는 **엔드포인트**를 복사해서 아래의 **YOUR_COG_ENDPOINT**에 붙여 넣는다.
3. 리소스에 있는 **위치**를 복사해서 아래의 **YOUR_COG_REGION**에 붙여 넣는다.
4. 아래 셀을 선택한 다음 셀 왼쪽에 있는 **셀 실행(>)** 버튼을 클릭하여 아래 코드를 실행한다.

In []:

```
cog_key = 'YOUR_COG_KEY'
cog_endpoint = 'YOUR_COG_ENDPOINT'
cog_region = 'YOUR_COG_REGION'

print('Ready to use cognitive services in {} using key {}'.format(cog_region, cog_key))
```

Cognitive 서비스 리소스에 있는 Speech 서비스를 이용하기 위해서는 Azure Cognitive Services Speech SDK를 설치해야 한다.

In []:

```
!pip install azure.cognitiveservices.speech
```

아래의 셀을 실행하여 오디오 파일로부터 음성을 텍스트로 바꾸고 이것을 Language Understanding 앱의 명령으로 사용해보자.

In []:

```
from python_code import luis
import os
import IPython
import os
from azure.cognitiveservices.speech import SpeechConfig, SpeechRecognizer, AudioConfig

try:

    # 오디오 파일로부터 음성 명령을 가져온다.
    file_name = 'light-on.wav'
    audio_file = os.path.join('data', 'luis', file_name)

    # 음성 인식기를 구성한다
    speech_config = SpeechConfig(cog_key, cog_region)
    audio_config = AudioConfig(filename=audio_file) # Use file instead of default (microphone)
    speech_recognizer = SpeechRecognizer(speech_config, audio_config)

    # 음성을 텍스트로 바꾸기 위해 일회성 및 동기화된 호출을 사용한다.
    speech = speech_recognizer.recognize_once()

    # 예측된 intent와 엔터티를 가져온다(python_code.home_auto.py에 있는 코드를 가져온다)
    action = luis.get_intent(luis_app_id, luis_key, luis_endpoint, speech.text)

    # 알맞은 이미지를 가져온다
    img_name = action + '.jpg'

    # 오디오를 실행하고 이미지를 보여준다
    IPython.display.display(IPython.display.Audio(audio_file, autoplay=True),
                           IPython.display.Image(data=os.path.join("data", "luis", img_name)))
except Exception as ex:
    print(ex)
```

위의 코드에서 음성 파일을 **light-off.wav**로 바꾸어 실행해보자.

심화학습

Language Understanding 에 대한 추가 문서는 다음을 참조하라 [service documentation](https://docs.microsoft.com/azure/cognitive-services/luis/) (<https://docs.microsoft.com/azure/cognitive-services/luis/>)



대화형 AI 살펴보기

600 XP



29분 • 학습 경로 • 모듈 1개

초급 AI 엔지니어 데이터 과학자 개발자 솔루션 아키텍처 학생 Bot Service

Cognitive Services

대화형 AI는 AI 에이전트와 인간 사용자 간의 대화를 처리하는 인공 지능 워크로드입니다.

필수 조건

Azure Portal 탐색 능력

 [책갈피](#) [컬렉션에 추가](#)

이 학습 경로의 모듈



QnA Maker 및 Azure Bot Service를 사용하여 봇 빌드

600 XP



29분 • 모듈 • 5 단위

4.7 (2,707)

봇은 여러 통신 채널을 통해 지원을 제공하는 인기 있는 방법입니다. 이 모듈에서는 QnA Maker 서비스와 Azure Bot Service를 사용하여 사용자 질문에 응답하는 봇을 만드는 방법을 설명합니다.

개요



✓ 600 XP ➔

QnA Maker 및 Azure Bot Service를 사용하여 봇 빌드

29분 • 모듈 • 5 단위

★★★★★ 4.7 (2,707)

평가하기

초급 AI 엔지니어 데이터 과학자 개발자 솔루션 아키텍처 학생

Bot Service Cognitive Services

봇은 여러 통신 채널을 통해 지원을 제공하는 인기 있는 방법입니다. 이 모듈에서는 QnA Maker 서비스와 Azure Bot Service를 사용하여 사용자 질문에 응답하는 봇을 만드는 방법을 설명합니다.

학습 목표

이 모듈을 완료하면 Azure Bot Service 봇을 사용하여 QnA Maker 기술 자료를 만들 수 있습니다.

📘 책갈피 Ⓛ 컬렉션에 추가

필수 조건

Azure Portal 탐색 능력

이 모듈은 이러한 학습 경로의 일부입니다.

[Azure Bot Service로 지능형 봇 만들기](#)

[대화형 AI 살펴보기](#)

소개

3분



QnA Maker 및 Azure Bot Service 시작

3분



연습 - 봇 만들기

20분



지식 점검

2분



요약

1분



100 XP



소개

3분

오늘날의 연결된 세상에서 사람들은 다양한 기술을 사용하여 의사 소통을 합니다. 예를 들면 다음과 같습니다.

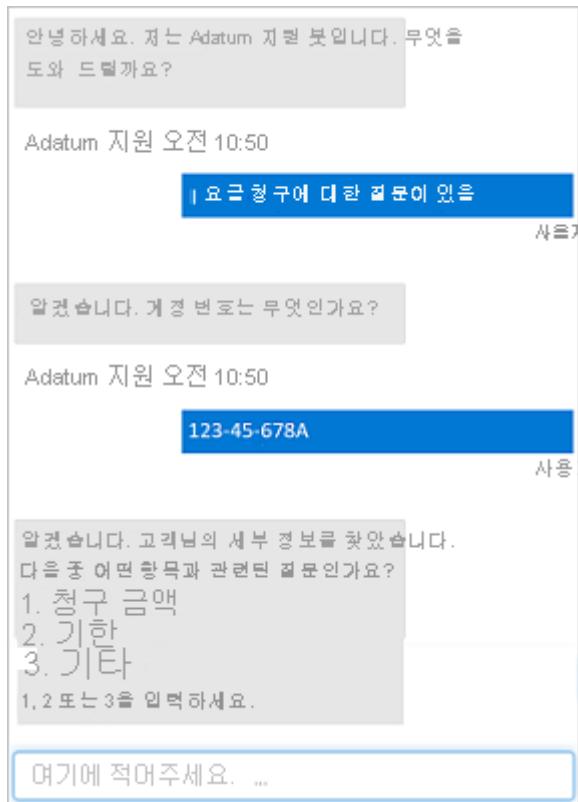
- 음성 통화
- 메시징 서비스
- 온라인 채팅 애플리케이션
- 메일
- 소셜 미디어 플랫폼
- 협업 작업 공간 도구

유비쿼터스 연결에 너무 익숙해져 있기 때문에 이미 사용하는 채널을 통해 원하는 조직에 쉽게 연락할 수 있고 해당 조직에서 즉시 응대할 것을 기대합니다. 또한 조직이 개별적으로 소통하면서 개인 차원의 복잡한 질문에 답변해 주기를 기대합니다.

대화형 AI

많은 조직에서 웹 브라우저 또는 전용 앱을 통해 액세스할 수 있는 FAQ(자주 묻는 질문)에 대한 지원 정보와 답변을 게시합니다. 조직이 제공하는 시스템과 서비스가 복잡하다면 특정 질문에 대한 답변을 찾기 어렵게 될 것입니다. 이러한 조직은 지원 담당자가 전화 통화, 메일, 문자 메시지, 소셜 미디어 및 기타 채널을 통한 지원 요청으로 업무에 과부하가 걸린 경우가 많습니다.

점점 더 많은 조직이 보통 '봇'이라고 하는 AI(인공 지능) 에이전트를 활용한 AI 솔루션으로 전환하여 소통에 사용되는 모든 채널을 통해 자동화된 일선 지원을 제공하고 있습니다. 봇은 채팅 인터페이스의 예제에서 보듯 대화 형식으로 사용자와 상호 작용하도록 설계되었습니다.



① 참고

여기에서 표시된 예제는 웹 사이트에서도 볼 수 있는 채팅 인터페이스입니다. 그러나 봇은 메일, 소셜 미디어 플랫폼, 심지어 음성 통화까지 포함한 여러 채널에서 작동하도록 설계될 수 있습니다. 사용되는 채널과 관계없이 봇은 일반적으로 자연어 및 사용자에게 해결책을 제시하는 제한된 옵션 응답의 조합을 사용하여 대화 흐름을 관리합니다.

대화는 일반적으로 차례로 교환되는 메시지의 형태를 취하며, 가장 일반적인 대화 교환 형식은 하나의 질문 뒤에 하나의 답변이 오는 형식입니다. 이 패턴은 많은 사용자 지원 봇의 기반을 구성하며 기존 FAQ 설명서를 기반으로 하는 경우가 많습니다. 이러한 종류의 솔루션을 구현하려면 다음이 필요합니다.

- 질문과 답변 쌍의 '기술 자료' - 일반적으로 일부 기본 제공 자연어 처리 모델을 사용하여 다양한 방식의 말로 표현된 질문을 동일한 의미론적 의미로 이해될 수 있도록 합니다.
- 하나 이상의 채널을 통해 기술 자료에 대한 인터페이스를 제공하는 '봇 서비스'.

다음 단원: QnA Maker 및 Azure Bot Service 시작

계속 >

< 이전

단위 2/5 ▾

다음 >

100 XP



QnA Maker 및 Azure Bot Service 시작

3분

다음 두 가지 핵심 기술을 조합하여 Microsoft Azure에서 사용자 지원 봇 솔루션을 쉽게 만들 수 있습니다.

- **QnA Maker.** 이 인지 서비스를 사용하면 기본 제공 자연어 처리 기능이 포함된 기술 자료를 만들고 게시할 수 있습니다.
- **Azure Bot Service.** 이 서비스는 Azure에서 봇을 개발, 게시 및 관리하기 위한 프레임워크를 제공합니다.

QnA Maker 기술 자료 만들기

사용자 지원 봇을 만드는 첫 번째 과제는 QnA Maker 서비스를 사용하여 기술 자료를 만드는 것입니다. 이 서비스는 기술 자료를 생성, 학습, 게시 및 관리하는 데 사용할 수 있는 전용 'QnA Maker 포털' 웹 기반 인터페이스를 제공합니다.

ⓘ 참고

QnA Maker REST API 또는 SDK를 사용하여 기술 자료를 만들고 관리하기 위한 코드를 작성할 수 있습니다. 그러나 대부분의 시나리오에서는 QnA Maker 포털을 사용하는 것이 더 쉽습니다.

QnA Maker Azure 리소스 프로비저닝

기술 자료를 만들려면 먼저 Azure 구독에서 QnA Maker 리소스를 먼저 프로비저닝해야 합니다. 기술 자료 작성을 시작하기 전에 Azure Portal에서 직접 이 작업을 수행하거나 QnA Maker 포털에서 기술 자료 개발을 시작하고 메시지가 표시될 때 리소스를 프로비저닝할 수 있습니다.

질문과 답변 정의

QnA Maker 리소스를 프로비저닝한 후 QnA Maker 포털을 사용하여 질문 및 답변 쌍으로 구성된 기술 자료를 만들 수 있습니다. 이러한 질문과 답변은 다음과 같을 수 있습니다.

- 기존 FAQ 문서 또는 웹 페이지에서 생성합니다.
- 미리 정의된 잡담 데이터 원본에서 가져옵니다.
- 수동으로 입력하고 편집합니다.

많은 경우에 위의 모든 기술을 조합해서 기술 자료를 만듭니다. 기존 FAQ 문서의 질문과 답변의 기본 데이터 세트부터 시작하여 잡담 원본에서 일반적인 대화 교환을 추가하고 추가 수동 입력 항목으로 기술 자료를 확장합니다.

기술 자료의 질문에는 동일한 의미의 질문을 통합하는 데 도움이 되는 '대체 어구'를 할당할 수 있습니다. 예를 들어 다음과 같은 질문을 포함할 수 있습니다.

'본사 위치는 어디입니까?'

다음과 같은 대체 어구를 추가하여 이 질문을 할 수 있는 다양한 방법을 예측할 수 있습니다.

'본사는 어디에 위치합니까?'

기술 자료 학습 및 테스트

질문 및 답변 쌍의 세트를 만든 후에는 기술 자료를 학습시켜야 합니다. 이 프로세스는 문자 그대로의 질문과 답변을 분석하고 기본 제공 자연어 처리 모델을 적용하여 질문의 정의에 명시된 정확한 문구가 아닌 경우에도 질문에 대해 적절한 답변을 매칭합니다.

학습 후 QnA Maker 포털의 기본 제공 테스트 인터페이스를 사용하여 질문을 제출하고 반환되는 답변을 검토하여 기술 자료를 테스트할 수 있습니다.

기술 자료 게시

학습된 기술 자료가 만족스럽다면 게시하여 클라이언트 애플리케이션에서 REST 인터페이스를 통해 사용하도록 할 수 있습니다. 기술 자료에 액세스하려면 클라이언트 애플리케이션에 다음이 필요합니다.

- 기술 자료 ID
- 기술 자료 엔드포인트
- 기술 자료 인증 키

Azure Bot Service를 사용하여 봇 빌드

기술 자료를 만들고 게시한 후 봇을 통해 사용자에게 제공할 수 있습니다.

기술 자료용 봇 만들기

Microsoft Bot Framework SDK를 사용하여 대화 흐름을 제어하고 QnA Maker 기술 자료와 통합하는 코드를 작성하면 사용자 지정 봇을 만들 수 있습니다. 그러나 더 쉬운 방법은 QnA Maker의 자동 봇 생성 기능을 사용하여 게시된 기술 자료용 봇을 만들고 몇 번의 클릭만으로 Azure Bot Service 애플리케이션으로 게시하는 것입니다.

봇 확장 및 구성

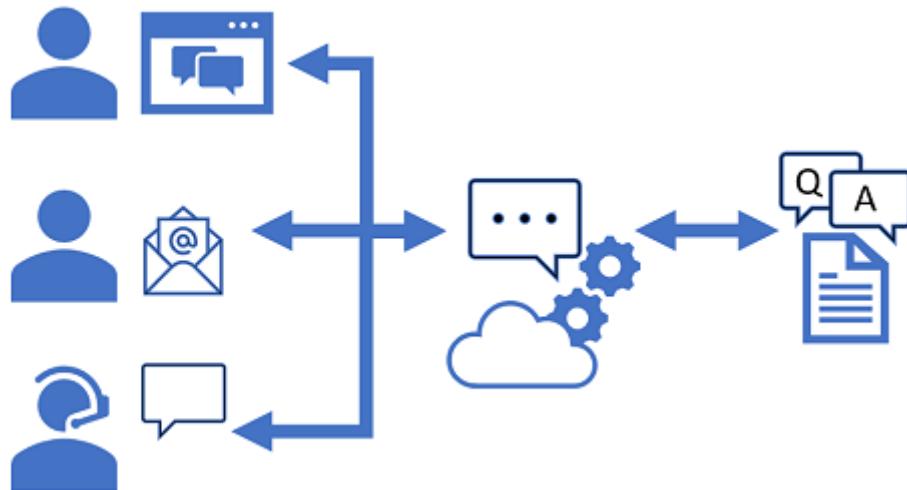
봇을 만든 후에는 Azure Portal에서 다음과 같이 관리할 수 있습니다.

- 사용자 지정 코드를 추가하여 봇의 기능을 확장합니다.
- 대화형 테스트 인터페이스에서 봇을 테스트합니다.
- 로깅, 분석 및 다른 서비스와의 통합을 구성합니다.

간단한 업데이트의 경우 Azure Portal에서 봇 코드를 직접 편집할 수 있습니다. 그러나 보다 포괄적인 사용자 지정을 원할 경우 소스 코드를 다운로드하여 로컬로 편집할 수 있습니다. 준비가 되면 봇을 Azure에 직접 다시 게시합니다.

채널 연결

봇을 사용자에게 제공할 준비가 되면 여러 '채널'에 연결할 수 있으므로 사용자가 웹 채팅, 메일, Microsoft Teams 및 기타 공통 커뮤니케이션 미디어를 통해 봇과 상호 작용할 수 있습니다.



사용자는 채널을 통해 봇에 질문을 제출하고 봇의 기반이 되는 기술 자료에서 적절한 답변을 받을 수 있습니다.

다음 단원: 연습 - 봇 만들기

계속 >

< 이전

단위 3/5 ▼

다음 >

✓ 100 XP



연습 - 봇 만들기

20분

QnA 서비스와 Azure Bot Service를 사용하여 봇을 빌드하는 방법을 배우는 가장 좋은 방법은 스스로 해보는 것입니다.

시작하기 전에

이 연습을 완료하려면 다음이 필요합니다.

- Microsoft Azure 구독. 구독이 아직 없다면 <https://azure.microsoft.com/free>에서 평가판에 가입할 수 있습니다.
- MicrosoftDocs/ai-fundamentals GitHub 리포지토리에서 Jupyter Notebook을 실행할 수 있는 Python 환경. GitHub의 랩 파일을 자신의 Python 환경(있는 경우)에 복제하거나, 아래 지침에 따라 Azure 구독에서 Azure Machine Learning 작업 영역을 만들 수 있습니다.

ⓘ 참고

이 모듈은 Azure Machine Learning 작업 영역을 사용하는 여러 가지 모듈 중 하나입니다. **Azure AI 기본 사항** 인증의 준비 과정에서 이 모듈을 완료하는 경우, 작업 영역을 한 번 만들고 해당 인증 트랙의 모든 모듈에 다시 사용할 수 있습니다. 연습을 완료한 후에는 정리 지침에 따라 모듈 간 컴퓨팅 리소스를 중지해야 합니다.

Azure Machine Learning 작업 영역 만들기

Azure 구독에 Azure Machine Learning 작업 영역이 이미 있는 경우 새 브라우저 탭에서 [Azure Machine Learning Studio](#)로 이동하고 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다. 없는 경우 다음 단계에 따라 새 작업 영역을 만듭니다.

1. Azure 구독과 연결된 Microsoft 계정을 사용하여 [Azure Portal](#)에 로그인합니다.
2. + 리소스 만들기를 선택하고, *Machine Learning*을 검색한 뒤 다음 설정을 사용하여 새 *Machine Learning* 리소스를 만듭니다.
 - **작업 영역 이름:** 원하는 고유한 이름을 입력
 - **구독:** 자신의 Azure 구독
 - **리소스 그룹:** 고유한 이름의 새 리소스 그룹 만들기
 - **위치:** 사용 가능한 위치 선택

3. 작업 영역 리소스가 만들어질 때까지 기다립니다(몇 분이 걸릴 수 있음). 그런 다음 포털로 이동하고 작업 영역의 **개요** 페이지에서 Azure Machine Learning Studio를 실행(또는 새 브라우저 탭을 열고 <https://ml.azure.com> 으로 이동)한 다음 Microsoft 계정을 사용하여 Azure Machine Learning Studio에 로그인합니다.
4. Azure Machine Learning Studio에서 왼쪽 상단에 있는 **☰** 아이콘을 설정/해제하면 인터페이스에 다양한 페이지가 표시됩니다. 해당 페이지를 사용하여 작업 영역에서 리소스를 관리할 수 있습니다.

컴퓨팅 인스턴스 만들기

이 연습에서 사용된 Notebook을 실행하려면 Azure Machine Learning 작업 영역에 컴퓨팅 인스턴스가 필요합니다. 컴퓨팅 인스턴스가 이미 있는 경우 시작하고, 없는 경우 다음 지침에 따라 새로 만듭니다.

1. [Azure Machine Learning Studio](#)에서 **관리** 아래에 있는 **컴퓨팅** 페이지를 확인합니다.
2. **컴퓨팅 인스턴스** 탭에서 다음 설정을 사용하여 새 컴퓨팅 인스턴스를 만듭니다.
 - **지역**: 사용 가능한 지역 선택
 - **가상 머신 유형**: CPU
 - **가상 머신 크기**: Standard_DS11_v2
 - **컴퓨팅 이름**: 고유한 이름 입력
3. 컴퓨팅 인스턴스가 시작될 때까지 기다립니다(1분 정도 걸릴 수 있음).

연습 파일 다운로드

이 모듈(및 기타 관련 모듈)에 사용되는 파일은 GitHub 리포지토리에 게시되는데, 이를 자신의 Python 환경에 복제해야 합니다. 이전 모듈에서 **ai-fundamentals** 리포지토리를 아직 복제하지 않은 경우 다음 단계를 사용하여 Azure Machine Learning 작업 영역에 복제합니다.

1. [Azure Machine Learning Studio](#)에서 **Notebooks** 페이지(**작성자** 아래)를 확인합니다. 이 페이지에는 Notebooks 실행에 사용할 수 있는 Notebook 편집기가 포함되어 있습니다.
2. **내 파일**에서 **□** 단추를 사용하여 다음 설정에 따라 새 파일을 만듭니다.
 - **파일 위치**: `Users/your user name`
 - **파일 이름**: Get-Files
 - **파일 형식**: Notebook
 - **이미 있는 경우 덮어쓰기**: 선택됨
3. 새 Notebook이 만들어지면 이전에 만든 컴퓨팅 인스턴스가 **컴퓨팅** 상자에서 선택되었는지, 그리고 상태가 **실행 중**인지 확인합니다. 그런 다음 Notebook에서 만든 사각형 셀에 다음 코드를 붙여넣습니다.

```
!git clone https://github.com/MicrosoftDocs/ai-fundamentals
```

4. 셀 옆에 있는 ▶ 단추를 사용하여 포함된 코드를 실행합니다. 이렇게 하면 GitHub에서 연습 파일이 복제됩니다.
5. 코드의 실행이 끝나고 파일의 체크아웃이 완료되면 내 파일 아래에 있는 ⌂ 단추를 사용하여 폴더 보기를 새로 고치고, **ai-fundamentals**라는 이름의 폴더가 만들어졌는지 확인합니다. 이 폴더에는 연습에 사용된 Notebook 및 기타 파일이 포함되어 있습니다.
6. Get-Files.ipynb Notebook 탭을 닫습니다.

연습 완료

Python 환경을 설정하고 **ai-fundamentals** 리포지토리를 복제했으면 봇 만들기를 실험할 준비가 된 것입니다.

1. **ai-fundamentals** 폴더에서 **QnA Bot.ipynb** Notebook을 엽니다. Azure Machine Learning Studio에서 Notebook 편집기를 사용하는 경우, <> 단추로 파일 탐색기 창을 축소하여 Notebook 탭에 집중할 수 있는 공간을 좀 더 마련합니다.
2. Notebook의 정보를 읽고 포함된 코드 셀을 순서대로 실행합니다.

정리

Azure Machine Learning Studio에서 컴퓨팅 인스턴스를 사용한 경우 Azure 크레딧을 불필요하게 사용하지 않도록 이를 중지해야 합니다.

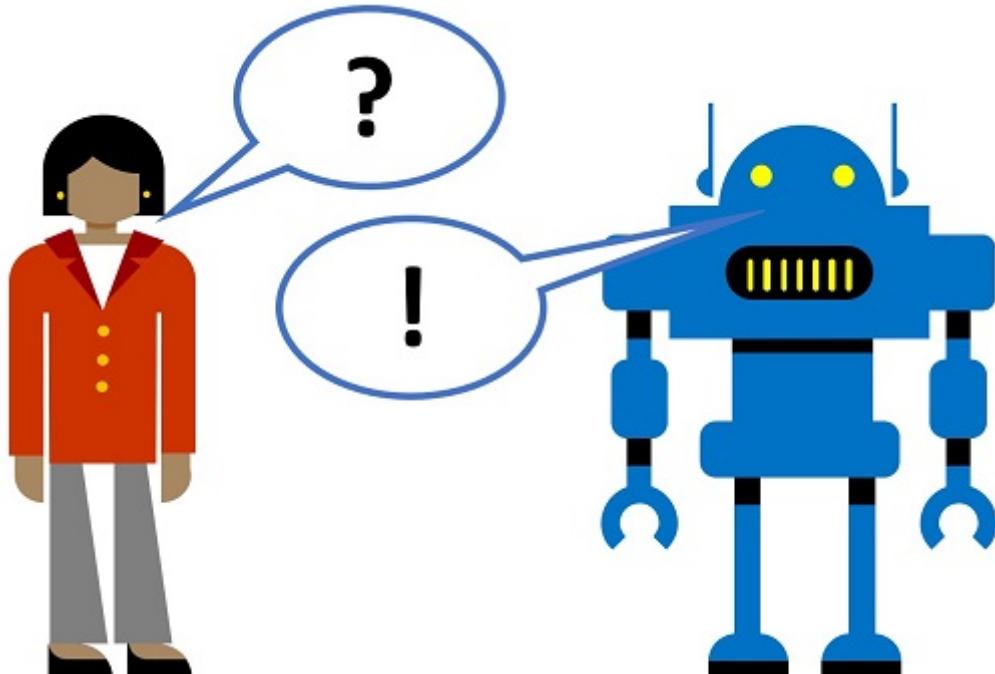
1. Azure Machine Learning Studio에서 관리 아래에 있는 컴퓨팅 페이지를 확인합니다.
2. 컴퓨팅 인스턴스 탭에서 컴퓨팅 인스턴스를 선택하고 중지 단추를 사용하여 중지합니다.

다음 단원: 지식 점검

계속 >

대화형(Conversational) AI

인스턴트 메시징, 소셜 미디어, 전자 메일 또는 기타 온라인 기술을 통해 다른 사람과 얼마나 자주 커뮤니케이션하는지 생각해 보라. 우리들 중 많은 사람들에게, 이것은 우리의 일반적인 연락방식이다. 회사에서 질문이 있을 때 모바일 장치에서 사용할 수 있는 대화 메시지를 사용하여 동료에게 보낼 수 있으므로 항상 연락할 수 있다.



봇은 이러한 종류의 채널을 사용하여 통신하는 AI 에이전트로, 소프트웨어 서비스와 자연스럽게 대화할 수 있다.

QnA Maker 기술 자료 만들기

고객 지원 시나리오의 경우, 웹 사이트의 채팅 창, 이메일 또는 음성 인터페이스를 통해 자주 묻는 질문을 해석하고 대답할 수 있는 봇을 만드는 것이 일반적이다. 봇 인터페이스의 기본에는 적절한 응답을 검색할 수 있는 질문과 적절한 대답을 제공하는 기술 자료(Knowledge Base)가 있다.

QnA Maker 서비스는 Azure에서 질의응답 쌍을 입력하거나 기존 문서나 웹 페이지에서 빠르게 지식 기반을 만들 수 있는 인지 서비스이다. 그런 다음 내장된 자연어 처리 기능을 사용하여 질문을 해석하고 적절한 답을 찾을 수 있다.

1. 브라우저의 새로운 탭을 열고 QnA Maker 포털(<https://qnamaker.ai>)로 (<https://qnamaker.ai>) 간다.
Azure 구독에 사용한 Microsoft 계정을 사용하여 로그인 한다.
2. QnA Maker 포털에서 **Create a knowledge base** 선택한다.
3. 만일 이전에 QnA서비스 리소스를 만든적이 없다면 **Create a QnA service**를 클릭한다. Azure 포털 서비스가 새로운 탭에서 실행되고 구독 서비스에서 QnA Maker 서비스를 만들 수 있다. 다음 내용으로 설정한다.
 - **이름:** QnA리소스를 위한 유일한 이름
 - **구독:** 현재 구독을 선택
 - **가격 책정 계층:** F0
 - **리소스 그룹:** Select an existing resource group or create a new one
 - **Azure Search 가격 계층:** F
 - **Azure Search 위치:** Any available location
 - **앱 이름:** 이름과 동일 ("azurewebsites.net" 이름이 자동으로 뒤에 붙는다)

- 웹 사이트 위치 : Azure Search 위치값과 동일하게 부여함
- App insights: 사용 안함

Note: 이미 무료 계층으로 QnA Maker나 Azure Search 리소스를 만들었다면 추가로 만들 수 없다. 그런 경우 F0 / F외에 다른 것을 선택해야 한다..

1. Azure portal에 있는 QnA 서비스와 관련 리소스들이 구축되기까지 기다린다.
2. QnA Maker 포털 탭으로 돌아와서 Step 2부분에서 Refresh 클릭하여 사용가능한 QnA서비스 리소스 목록이 나타나도록 한다.
3. 다음과 같은 순서로 QnA서비스를 기술자료(KB)에 연결하도록 한다.
 - Microsoft Azure Directory ID: 구독에 대한 Azure 디렉터리 ID(보통 기본 디렉터리)
 - Azure subscription name: Azure 구독을 선택
 - Azure QnA service: 이번 과정에서 생성한 QnA 서비스 리소스를 선택
 - Language: English

팁: 만일 이런 내용들이 모두 나타나지 않는다면 QnA Maker 포털 브라우저 페이지를 새로고침한다.

1. Step 3 부분에서, Margie's Travel KB라고 이름란에 입력한다.
2. Step 4 부분에서, URL 상자에, [*https://github.com/MicrosoftDocs/ai-fundamentals/raw/master/data/qna_bot/margies_faq.docx*](https://github.com/MicrosoftDocs/ai-fundamentals/raw/master/data/qna_bot/margies_faq.docx)()를 입력하고 + Add URL 을 클릭한다. 그런 다음 Chit-chat 밑에서 Professional 을 선택한다.
3. Step 5 부분에서, Create your KB을 클릭한다.
4. 기술자료가 생성될 때까지 1분 이상 기다린다. 그리고 나서 FAQ 문서로부터 가져온 질문과 답변 내용과 미리 정한 chit-chat 응답 내용을 확인한다.

기술 자료 편집하기

기술 자료는 FAQ 문서의 세부 정보와 몇 가지 사전 정의된 응답을 기반으로 한다. 사용자 지정 질문과 대답 쌍을 추가하여 이러한 쌍을 보완할 수 있다.

1. + Add QnA pair 클릭한다.
2. Question 상자에, Hello 을 입력하고 + Add alternative phrasing 선택한 다음 Hi 입력한다.
3. Answer 상자에, Hello 를 입력한다.

기술 자료 학습시키기 및 테스트하기

이제 기술 자료가 완성되어 QnA Maker 포털에서 테스트 할 수 있다.

1. 페이지의 위쪽 오른편에서 Save and train 를 클릭하여 기술 자료를 학습시킨다.
2. 학습이 끝나면 ← Test 를 클릭하여 테스트 창을 연다.
3. 테스트 창에서, 밑에다 Hi라는 메시지를 입력하고 엔터를 누른다. 그러면 Hello라는 응답을 가져와서 보여 주게 될 것이다.
4. 테스트 창에서, 밑에다 I want to book a flight라는 메시지를 입력하고 엔터를 누른다. FAQ로부터 알맞은 메시지가 되돌아 오는지 확인해보자.
5. 기술 자료에 대한 테스트를 마쳤다면 → Test 클릭하여 테스트 창을 닫는다.

기술 자료를 사용하여 봇(Bot)생성하기

기술 자료는 클라이언트 응용 프로그램이 일종의 사용자 인터페이스를 통해 질문에 대답하는 데 사용할 수 있는 백엔드 서비스를 제공한다. 일반적으로 이러한 클라이언트 응용 프로그램은 봇이다. 봇에서 기술 자료를 사용하려면 HTTP를 통해 액세스할 수 있는 서비스로 게시해야 한다. 그런 다음 Azure Bot Service를 사용하여 기술 자료를 사용하여 사용자 질문에 답변하는 봇을 만들고 호스팅할 수 있다.

1. QnA Maker 페이지의 위에서 **Publish**를 클릭한다. 그리고 나서 **Margie's Travel KB** 페이지에서 **Publish**를 클릭한다.
2. 서비스가 구축된 후에 **Create Bot**을 클릭한다. 새로운 탭에서 Azure 포털이 열려서 Azure 구독에서 Web App Bot을 하나 만든다.
3. Azure 포털에서, 다음과 같은 설정사항(대부분은 이미 만들어져 있음)으로 웹 앱 봇(Web App Bot)을 만든다
 - **봇 핸들**: 봇에 대한 유일한 이름
 - **구독**: Azure 구독 선택
 - **리소스 그룹**: QnA Maker 리소스를 포함하고 있는 리소스 그룹
 - **위치**: QnA Maker 서비스와 같은 위치.
 - **가격 책정 계층**: F0
 - **앱 이름**: 봇 핸들과 동일한 이름으로 .azurewebsites.net이 자동으로 뒤에 붙음
 - **SDK 언어**: C# 또는 Node.js를 선택
 - **QnA 인증 키**: *이것은 자동으로 QnA 기술자료에 설정되거나 인증키를 사용해야 한다.
 - **앱 서비스 계획/위치**: 이것은 자동으로 알맞은 계획과 위치가 설정되도록 두어야 한다
 - **Application Insights**: Off
 - **Microsoft 앱 ID 및 암호**: 자동으로 App ID와 암호가 생성된다.
4. 봇이 만들어질 때까지 기다린다(윗쪽 오른편에 알림아이콘에 벨 모양으로 동작으로 알림이 나타난다) 그리고 나서 구축이 완료된 후 **리소스로 이동**(또는 홈 페이지에서 **리소스 그룹**을 클릭한 다음 웹 앱 봇이 생성된 리소스 그룹을 선택한다)을 클릭한다.
5. 봇 페이지에서 **Test in Web Chat** 메뉴를 선택하고 **Hello and welcome!** 메시지를 출력할 때까지 기다린다(초기화 되려면 몇초를 기다려야 할 수 있다).
6. 테스트 챗 인터페이스를 사용하여 기술 자료에 있는 질문을 입력해서 어떻게 나오는지 확인한다. 예를 들면 *I need to cancel my hotel*이라 입력한다.

Access the Bot through a Channel

봇을 사용하여 하나 이상의 채널을 통해 사용자를 위한 인터페이스를 제공할 수 있다. 예를 들어, 동일한 봇이 웹 채팅 인터페이스, 이메일 및 Microsoft Teams를 통한 상호 작용을 지원할 수 있다.

1. Azure 포털에서, 만든 봇 서비스를 선택하고 **Channels** 페이지를 확인한다.
2. **Web Chat** 채널이 자동으로 추가된 것을 확인하고 일반적인 의사소통 플랫폼을 위한 사용 가능한 다른 채널들이 나타나 있다
3. **Web Chat** 채널 업에서 **편집**을 클릭한다. 이렇게 하면 웹 페이지에 봇을 추가하는데 필요한 설정들이 나타나 있다. 봇을 임베드하기 위해서는 봇을 위해 생성된 암호키와 HTML 임베드 코드가 필요하다.
4. **임베드 코드**를 복사하고 아래 셀의 "란에 붙여 넣는다."
5. 암호키들 중 하나를 선택하고 **표시**(어떤 것이든지 상관 없음)를 클릭한 후에 값을 복사 한다. 그리고 나서 아래의 HTML 임베드 코드에 있는 YOUR_SECRET_HERE를 대체하여 붙여 넣는다.
6. HTML 코드에 있는 **min-height** 값을 **200px**로 바꾼다(기본 값인 500px 대신 넣어야 함). 이것은 마우스를 스크롤 할 필요 없이 HTML 페이지를 볼 수 있다. This will help ensure the HTML interface is visible without scrolling.
7. 아래 셀의 왼쪽에 있는 **Run cell (▷)** 버튼을 클릭하여 HTML 코드를 랜더링한다.
8. HTML 채팅 인터페이스에서 *Who is Margie?*나 *What destinations can I go to?*와 같은 질문을 넣어서 테스트해보자.

In [2]:

```
%%html
```

```
<!-- EMBED CODE GOES HERE -->
```

봇으로 실험해보라. FAQ의 질문에 대해 매우 정확하게 답변할 수 있지만 교육받지 않은 질문을 해석하는 능력은 제한적이다. 언제든지 QnA Maker 포털을 사용하여 기술 자료를 편집하여 개선한 후 다시 게시할 수 있다.

심화학습

- QnA Maker 서비스에 대해서 더 알아보기 위해서는 [QnA Maker 문서](https://docs.microsoft.com/azure/cognitive-services/qnamaker/) (<https://docs.microsoft.com/azure/cognitive-services/qnamaker/>)를 확인하라.
- Microsoft Bot 서비스에 대해서 더 알아보기 위해서는 [Azure Bot 페이지](https://azure.microsoft.com/services/bot-service/) (<https://azure.microsoft.com/services/bot-service/>)를 확인하라.