MathWorks®

머신 러닝 시작하기

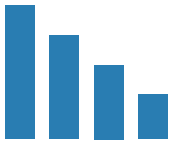
한번에 수행되지 않는 머신 러닝

머신 러닝에서는 시작부터 완료까지 한번에 수행되는 경우는 거의 없으며, 계속해서 새로운 방법을 반복적으로 시도하게 됩니다. 이 장에서는 체계적인 머신러닝 워크플로우를 설명하고 각 주요 작업에 대해 살펴봅니다.

머신 러닝 과제

대부분의 머신 러닝 과제는 데이터 처리 및 적합한 모델 찾기에 관련됩니다.

데이터의 형태와 크기는 다양합니다. 실제 데이터셋은 정리되지 않고, 불완전하며, 형식이 다양할 수 있습니다. 간단한 숫자 데이터만 있을 수도 있지만 경우에 따라 센서 신호, 텍스트, 카메라의 이미지 스트리밍 같은 여러 데이터 유형을 결합할 수도 있습니다.



데이터 전처리에는 전문 지식과 도구가 필요할 수 있습니다. 예를 들어 객체 감지 알고리즘을 훈련할 특징을 선택하려면 이미지 처리에 대한 전문 지식이 필요합니다. 데이터 유형에 따라 다른 전처리 접근법이 필요합니다.

데이터를 피팅할 최적 모델을 찾는 데는 시간이 걸립니다. 적합한 모델을 선택하는 것은 균형을 잡는 작업입니다. 유연성이 높은 모델은 노이즈일 수 있는 사소한 차이를 모델링하여 데이터를 오버피팅하는 경향이 있습니다. 반면, 단순 모델은 너무 많은 가정을 포함할 수 있습니다. 모델 속도, 정확성 및 복잡성 간에는 항상 상충관계가 존재합니다.

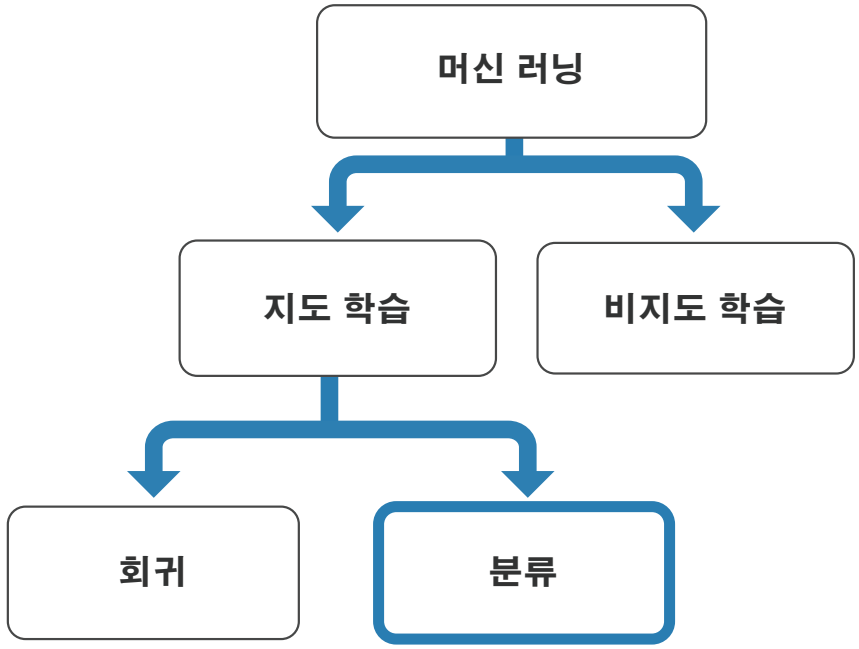
너무 어려운 작업처럼 들리십니까? 낙담하지 마십시오. 시행착오가 머신 러닝의 핵심이라는 점을 기억하십시오. 한 접근법이나 알고리즘이 작동하지 않으면 다른 방법을 시도하면 됩니다. 하지만 체계적인 워크플로가 있다면 순조롭게 출발할 수 있습니다.

시작하기 전에 고려할 질문

모든 머신 러닝 워크플로우는 세 가지 질문으로 시작됩니다.

- 어떤 유형의 데이터를 사용하고 있습니까?
- 데이터에서 어떤 통찰력을 얻고 싶습니까?
- 그러한 통찰력을 어떻게, 어디에 적용하겠습니까?

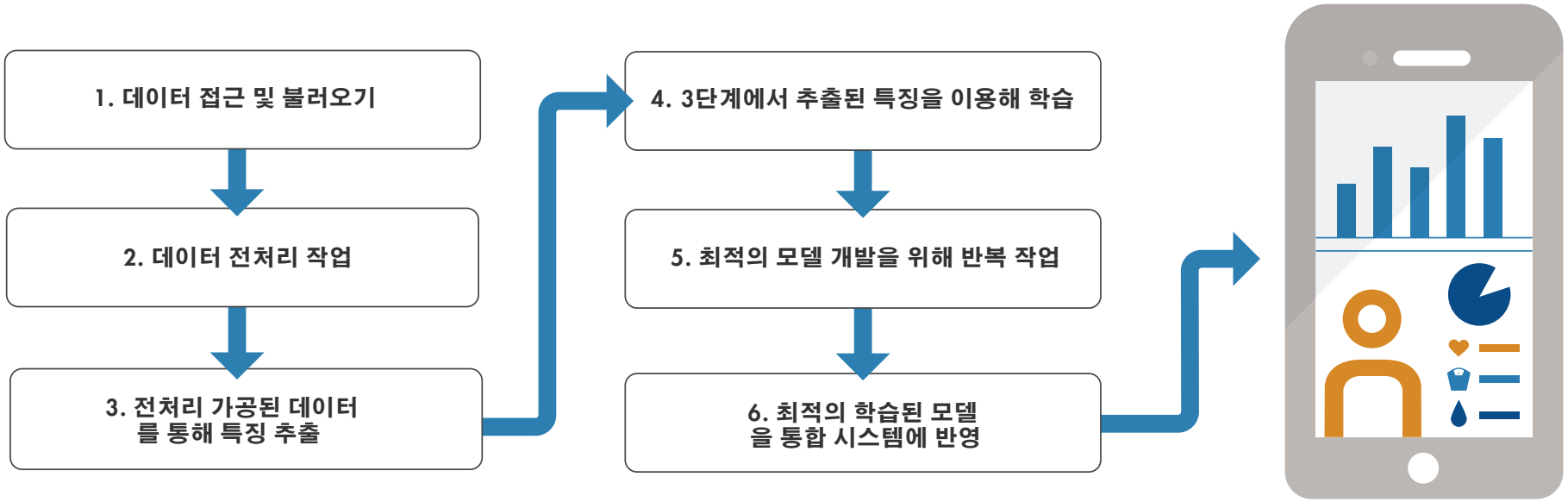
이러한 질문에 대한 답변을 통해 지도 학습을 사용할지 또는 비지도학습을 사용할지 결정할 수 있습니다.



온도나 주가 같은 연속 변수의 미래 가치 등에 대한 예측을 작성하거나 웹캠 비디오 화면에서 자동차 제조업체를 식별하는 등의 분류를 수행하도록 모델을 훈련해야 할 경우 지도 학습을 선택합니다.

데이터를 탐색해야 하고 데이터를 클러스터로 분할하는 것과 같이 좋은 내부 표현을 찾도록 모델을 훈련하려는 경우 비지도학습 선택합니다.

워크플로우 한눈에 보기



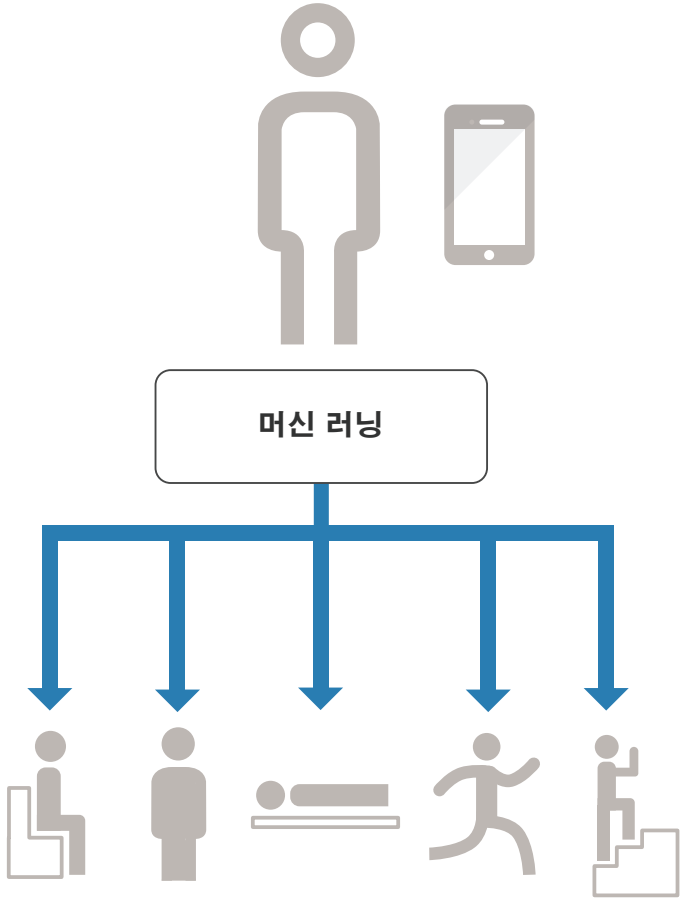
다음 섹션에서는 실례로 건강 상태 모니터링 앱을 사용하여 단계를 자세히 살펴보겠습니다. 전체 워크플로는 MATLAB®에서 완료됩니다.

신체활동 분류기 개발을 위한 학습

이 예제는 휴대 전화의 건강 상태 모니터링 앱 개발을 위한 예제입니다. 입력 데이터는 3차원 위치 및 가속 센서 데이터로 구성됩니다. 출력은 걷기, 서기, 달리기,계단 오르기, 눕기로 구성되었습니다.

입력 데이터를 이용해 신체 활동 분류기 모델 개발을 위해 학습을 수행합니다. 여기서는 분류기 개발이 목적이기 때문에 지도 학습을 적용하겠습니다.

개발된 분류기는 App개발에 활용되어 매일 사용자의 신체 활동을 추적할 수 있습니다.



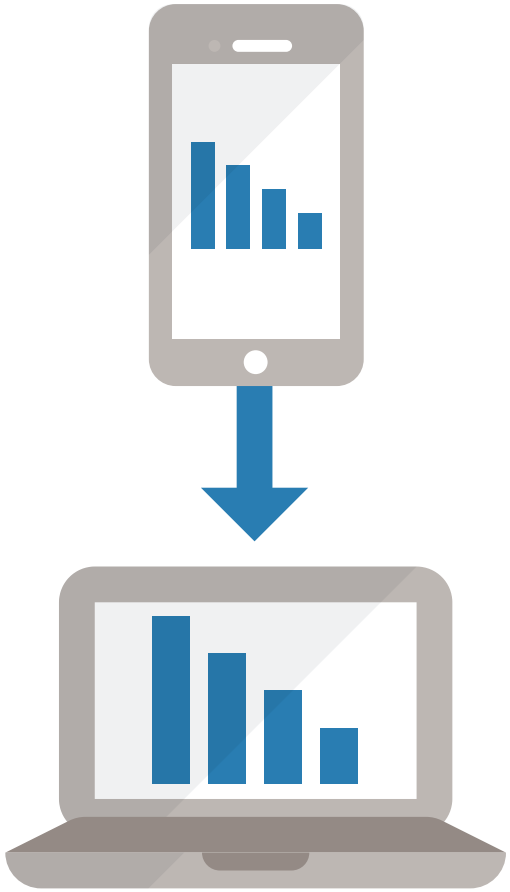
1 1단계: 데이터 로드

위치 및 가속도 센서 데이터를 불러오기 위해 다음을 수행합니다.

1. 전화기를 들고 앉아 전화기의 데이터를 기록하고 "앉기"라는 텍스트 파일에 저장합니다.
2. 전화기를 들고 서서 전화기의 데이터를 기록하고 "서기"라는 두 번째 텍스트 파일에 저장합니다.
3. 분류할 각 활동에 대한 데이터를 수집할 때까지 단계를 반복합니다.

레이블이 지정된 데이터 세트를 텍스트 파일에 저장합니다. 텍스트나 CSV 등의 플랫 파일 형식은 쉽게 작업하고 데이터를 간단히 가져올 수 있습니다.

머신 러닝 알고리즘은 노이즈와 중요한 정보의 차이를 알려줄 만큼 스마트하지 않습니다. 훈련에 데이터를 사용하기 전에 데이터가 정리되고 완전한지 확인해야 합니다.



2 2단계: 데이터 전처리

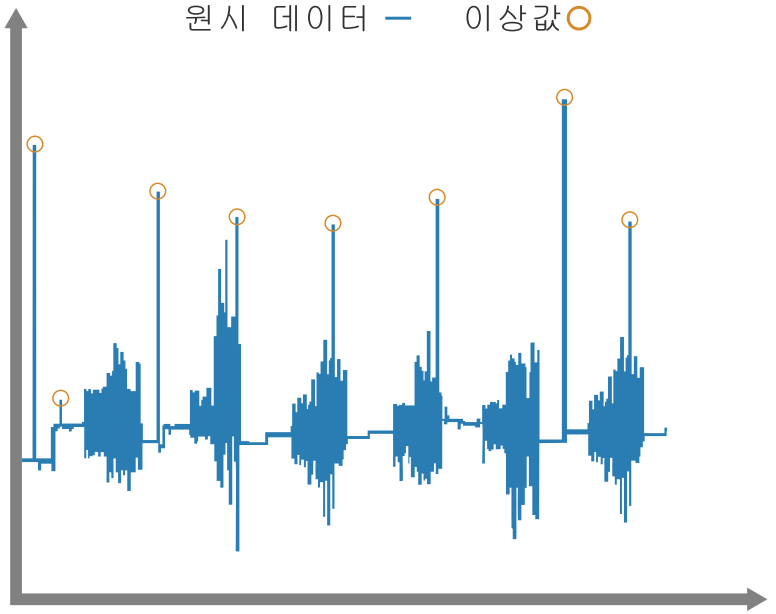
데이터를 MATLAB로 가져오고 각각 레이블이 지정된 세트를 플로팅합니다.
데이터를 전처리하려면 다음을 수행합니다.

1. 나머지 데이터 외부에 있는 이상값 데이터 포인트를 검색합니다.

이상값을 무시할지 여부 또는 이상값이 모델에서 고려해야 할 현상을 나타내는지 여부를 결정해야 합니다. 이 예제에서는 이상값을 무시해도 됩니다(데이터를 기록하는 동안 의도치 않게 이동한 것으로 확인됨).

2. 누락된 값이 있는지 확인합니다(기록 중에 연결이 끊어져 데이터가 손실되었을 수 있기 때문).

누락된 값을 그냥 무시할 수 있지만 이렇게 하면 데이터 세트 크기가 감소합니다. 또는 다른 샘플의 비슷한 데이터를 보간하거나 사용하여 누락된 값을 근삿값으로 대체할 수 있습니다.



활동 추적 데이터의 이상값.

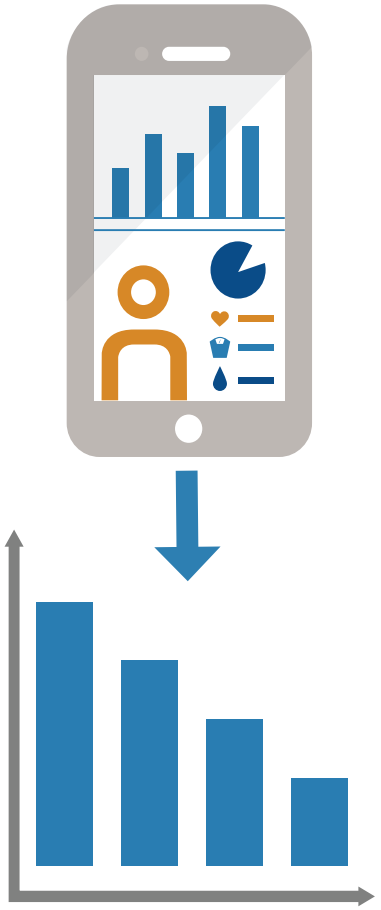
많은 응용프로그램에서 이상값은 중요한 정보를 제공합니다. 예를 들어 신용 카드 위조 감지 앱에서 이상값은 고객의 일상적인 구매 패턴을 벗어나는 구매를 나타냅니다.

2 2단계: 데이터 전처리 *계속*

3. 알고리즘이 전화기의 이동이 아니라 주체의 이동에 초점을 맞추도록 가속도계 데이터에서 중력 효과를 제거합니다. 일반적으로 바이쿼드 필터와 같은 간단한 하이패스 필터가 이를 위해 사용됩니다.

4. 데이터를 두 개의 세트로 나눕니다. 일부 데이터는 테스트용으로 저장하고(테스트 세트) 나머지는 모델 작성에 사용합니다(훈련 세트). 이를 홀드아웃이라고 하며 유용한 교차 유효성 검사 기법입니다.

모델링 프로세스에 사용되지 않은 데이터를 대상으로 모델을 테스트하여 알 수 없는 데이터에서는 모델이 어떻게 작동하는지 확인합니다.



3 3단계: 특징 도출

특징 도출(특징 엔지니어링 또는 특징 추출이라고도 함)은 머신 러닝의 가장 중요한 부분 중 하나입니다. 특징 도출은 원시 데이터를 머신 러닝 알고리즘에서 사용할 수 있는 정보로 전환합니다.

활동 추적기의 경우 가속도계 데이터의 빈도 콘텐츠를 캡처하는 특징을 추출하고자 합니다. 알고리즘에서는 이러한 특징을 기반으로 걷기(낮은 빈도)와 달리기(높은 빈도)를 구분할 수 있습니다. 선택한 특징이 포함된 새 테이블을 만듭니다.

특징 선택을 사용하여 다음을 수행합니다.

- 기계 학습 알고리즘의 정확도 개선
- 고차원 데이터 세트의 모델 성능 향상
- 모델 해석 가능성을 향상
- 오버피팅 방지

3 3단계: 특징 도출 *계속*

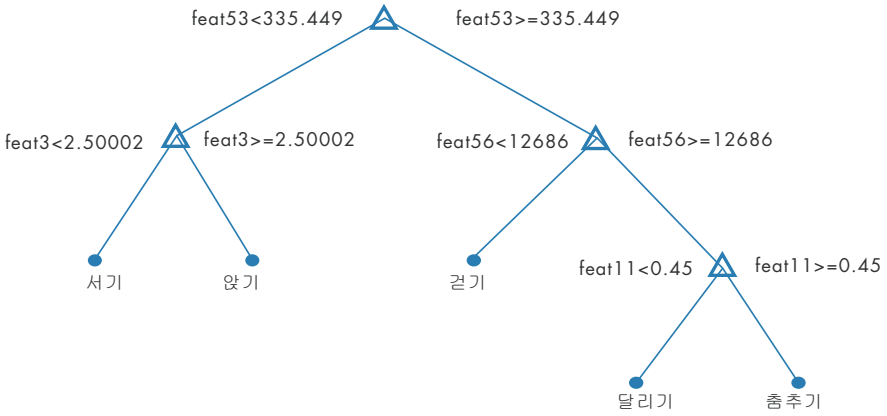
도출할 수 있는 특징의 수에는 제한이 없습니다. 하지만 다양한 유형의 데이터에 일반적으로 사용되는 많은 기법이 있습니다.

데이터 유형	특징 선택 작업	기법
센서 데이터	원시 센서 데이터에서 신호 속성을 추출하여 하이 레벨 정보 생성	**최대값 분석** – FFT를 수행하고 상위(Dominant) 빈도 식별 **펄스 및 전환 메트릭** – 상승 시간, 하강 시간, 정착 시간 등의 신호 특성 도출 **스펙트럼 측정** – 신호 출력, 대역폭, 중심 주파수, 중간 주파수 플로팅
이미지 및 비디오 데이터	모서리 위치, 해상도, 색상 등의 특징 추출	**시각적 단어 모음** – 모서리, 코너, 색상 영역 등의 로컬 이미지 특징에 대한 히스토그램 작성 **HOG(Histogram of Oriented Gradients)** – 로컬 그라디언트 방향에 대한 히스토그램 작성 **최소 고유값 알고리즘** – 이미지에서 코너 위치 감지 **모서리 감지** – 밝기가 급격히 바뀌는 지점 식별
트랜잭션 데이터	데이터에서 정보를 개선하는 도출된 값 계산	**타임스탬프 분해** – 타임스탬프를 일, 월 등의 성분으로 분해 **집계 가치 계산** – 특정 이벤트가 발생한 총 횟수 등의 상위 특징 작성

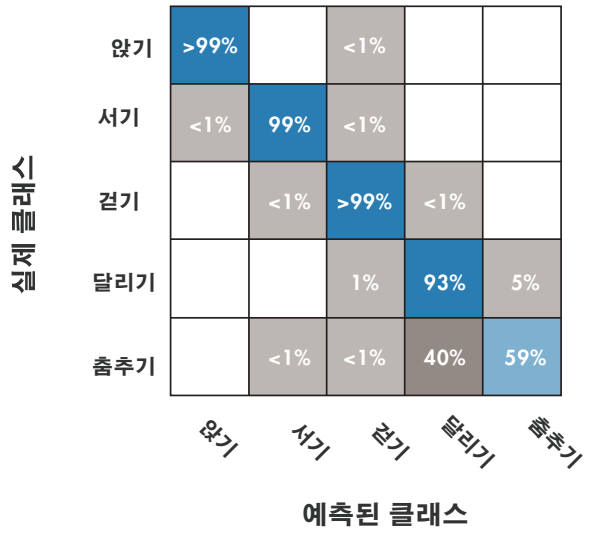
4 4단계: 모델 작성 및 훈련

모델 작성 시 간단한 것부터 시작하는 것이 좋습니다. 간단해야 더 빠르게 실행하고 더 쉽게 해석할 수 있습니다.

기본 의사결정 트리에서 시작합니다.



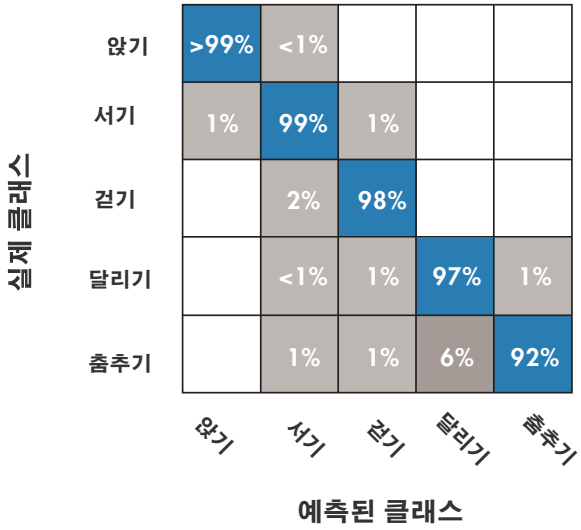
얼마나 잘 작동하는지 확인하기 위해 1단계에서 작성한 실제 클래스 레이블과 모델별로 수행된 분류를 비교하는 테이블인 오차 행렬을 플로팅합니다.



오차 행렬에 의하면 모델에 춤추기와 달리기를 구분하는 데 문제가 있음을 알 수 있습니다. 이 유형의 데이터에는 의사결정 트리가 작동하지 않을 수 있습니다. 몇 가지 다른 알고리즘을 시도해 보겠습니다.

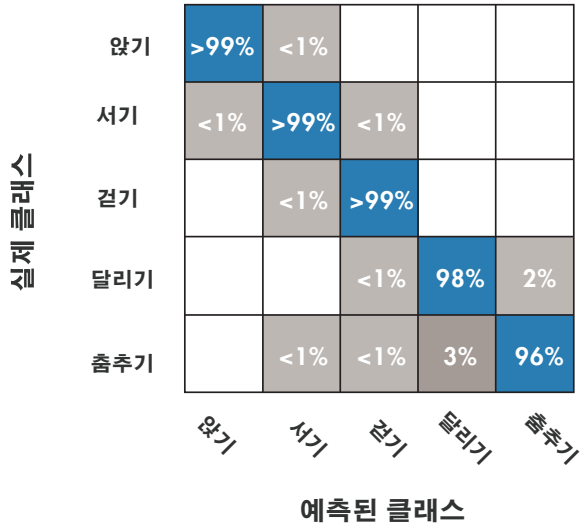
4 4단계: 모델 작성 및 훈련 *계속*

모든 훈련 데이터를 저장하고, 새 포인트를 훈련 데이터에 비교하고, “K” 최근접 포인트의 가장 빈번한 클래스를 반환하는 단순 알고리즘인 KNN(K-Nearest Neighbor)에서 시작합니다. 단순 의사결정 트리의 94.1% 정확도에 비해 이 모델은 98% 정확도를 제공합니다. 오차 행렬도 향상된 모습을 보입니다.



하지만 KNN은 예측을 하는 데 모든 훈련 데이터가 필요하므로 실행 시 상당한 양의 메모리가 사용됩니다.

여기서는 선형 판별 모델을 시도하지만 결과가 향상되지 않습니다. 마지막으로 다계층 SVM(서포트 벡터 머신)을 시도합니다. 다음과 같이 SVM이 매우 잘 작동하고 이제 99% 정확도를 얻습니다.



모델에 반복 수행하고 여러 알고리즘을 시도하여 목표를 달성했습니다. 이제 분류기에서 여전히 춤추기와 달리기를 안정적으로 구별할 수 없는 경우 모델을 개선할 여러 방법을 살펴보겠습니다.

5 5단계: 모델 개선

모델 개선에는 모델을 더 간단히 만들거나 복잡성을 추가하는 두 가지 방향이 있습니다.

간소화

먼저 특징 수를 줄일 수 있는 방법을 찾아봅니다. 널리 사용되는 특징 감소 기법으로는 다음이 있습니다.

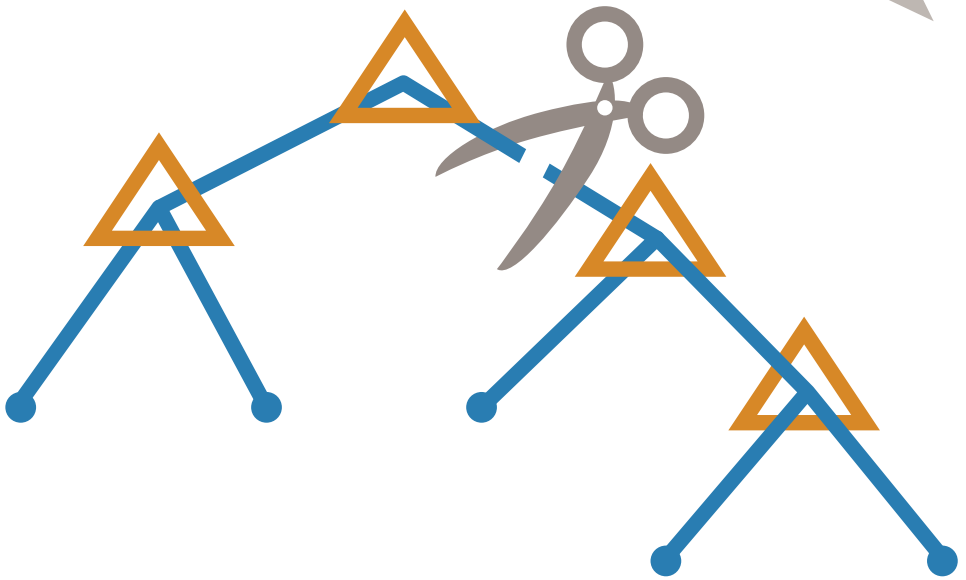
- 상관관계 행렬 – 상관관계가 깊지 않은 변수(또는 특징)를 제거할 수 있도록 변수 간 관계를 표시합니다.
- PCA(주성분 분석) – 원래 특징 간에 주요 차이를 캡처하는 특징 조합을 찾아 중복성을 제거하고 데이터셋에서 강력한 패턴을 도출합니다.
- Sequential Feature selection기법을 이용한 변수 축약 – 더 이상 성능 향상이 없을 때까지 모델에서 반복해서 변수를 줄입니다.

다음으로 모델 자체를 줄이는 여러 방법을 살펴봅니다. 다음과 같은 방법이 있습니다.

- 의사결정 트리에서 분기 정리
- 앙상블 학습을 통해 모델의 불필요한 변수를 줄입니다.

좋은 모델에는 가장 예측 가능성이 높은 특징만 포함됩니다. 잘 일반화되는 단순 모델이 새 데이터에 대해 일반화하거나 잘 훈련될 수 없는 복잡한 모델보다 효율적입니다.

머신 러닝의 경우 기타 여러 계산 프로세스에서처럼 모델을 간소화하여 더 이해하기 쉽고, 더 강력하고, 더 계산 효율적인 모델을 만들 수 있습니다.



5 5단계: 모델 개선 *계속*

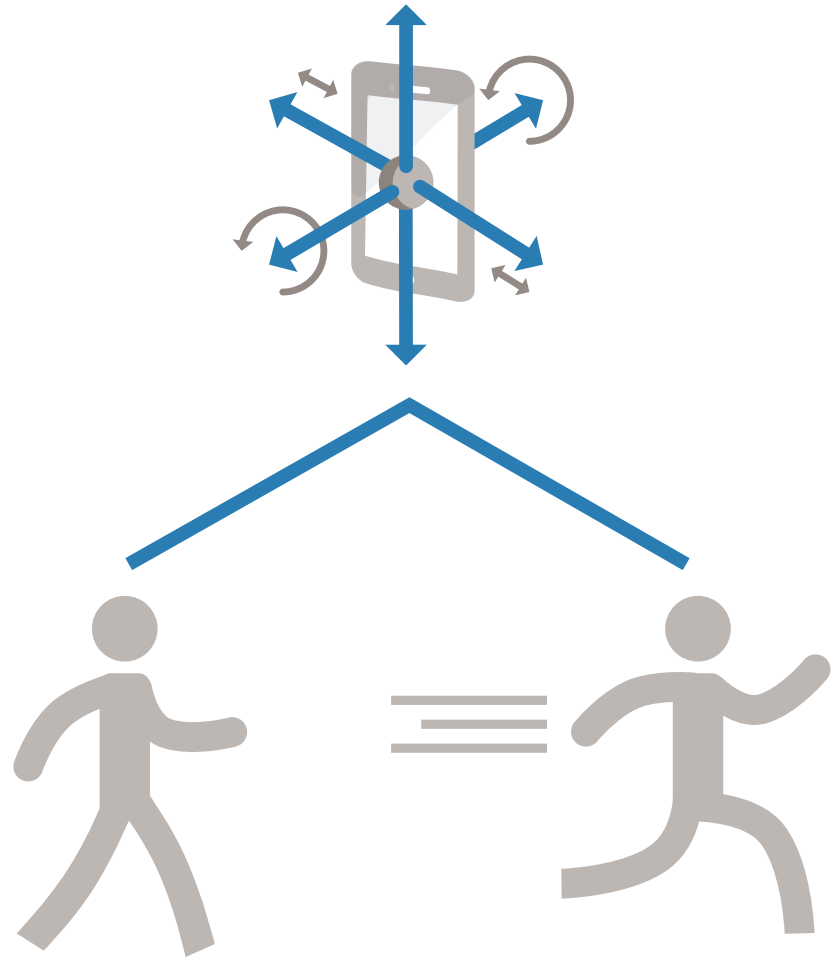
복잡성 추가

지나친 일반화로 인해 모델에서 춤추기와 달리기를 구별할 수 없다면 모델을 더 미세 조정하는 방법을 찾아야 합니다. 방법은 다음과 같습니다.

- 모델 조합 사용 – 간단한 여러 모델을 해당 모델이 단독으로 할 수 있는 것보다 데이터에서 추세를 더 잘 나타낼 수 있는 더 큰 모델로 병합합니다.
- 더 많은 데이터 추가 - 가속도 센서와 위치 센서 데이터를 살펴봅니다. 위치 센서 데이터는 활동 중에 휴대 전화의 방향을 기록합니다. 이 데이터는 여러 가지 활동에 대한 고유한 특징을 제공할 수 있습니다. 예를 들어 달리기에는 고유한 가속도 및 회전 조합이 있을 수 있습니다.

모델을 조정한 후 전처리 중에 준비한 테스트 데이터에 대해 성능을 검증합니다.

모델이 테스트 데이터에서 활동을 안정적으로 분류할 수 있다면 모델을 전화기로 이동하고 추적을 시작할 준비가 된 것입니다.



추가 정보

자세히 살펴볼 준비가 되셨습니까? 다음 리소스를 통해 머신 러닝 방법, 예제, 도구를 자세히 알아보십시오.

시청 자료

머신 러닝으로 간편하게 34:34

센서 데이터 분석을 위한 신호 처리 및 머신 러닝 기법 42:45

읽기 자료

지도학습 워크플로우 및 알고리즘

MATLAB 분석을 사용한 데이터 기반 통찰력: 에너지 부하 예측 사례 연구

살펴보기

MATLAB 머신 러닝 예제

Classification Learner 앱을 사용한 데이터 분류

93112v00