Rozdział 2

Rozdział 2

2.1 Czym jest uczenie maszynowe?

Uczenie maszynowe (ang. Machine Learning, ML) to poddziedzina sztucznej inteligencji, której celem jest tworzenie algorytmów i modeli zdolnych do samodzielnego doskonalenia swoich działań na podstawie analizy danych.

Zamiast być bezpośrednio programowane do wykonywania określonych zadań, systemy uczące się "uczą się" na podstawie danych, aby dokonywać przewidywań, rozpoznawać wzorce lub podejmować decyzje. Podstawową różnicą między klasycznym programowaniem a uczeniem maszynowym jest sposób, w jaki tworzony jest system.

W tradycyjnym programowaniu programista definiuje każdy krok algorytmu, podczas gdy w uczeniu maszynowym model uczy się, jak wykonać zadanie na podstawie dostarczonych danych. Uczenie maszynowe odgrywa kluczową rolę w wielu nowoczesnych technologiach.

Przykłady obejmują rozpoznawanie mowy, przetwarzanie języka naturalnego, systemy rekomendacyjne, rozpoznawanie obrazów i autonomiczne pojazdy.

Dzięki zdolnościom do adaptacji i nauki z nowych danych, systemy ML mogą dostosowywać się do zmieniających się warunków i poprawiać swoje wyniki w czasie.

Uczenie maszynowe można podzielić na różne kategorie w zależności od sposobu, w jaki model uczy się na podstawie danych oraz rodzaju zadań, które ma wykonać.

2.2 Typy uczenia maszynowego Uczenie maszynowe można ogólnie podzielić na trzy główne kategorie:
uczenie nadzorowane, uczenie nienadzorowane oraz uczenie przez wzmacnianie.

Każdy z tych typów ma swoje specyficzne zastosowania i jest używany do rozwiązywania różnych rodzajów problemów.
Uczenie nadzorowane (Supervised Learning) Uczenie nadzorowane to najczęściej stosowany typ uczenia maszynowego.

Polega ono na trenowaniu modelu na zbiorze danych, w którym każdemu przykładowi odpowiada znana etykieta lub wynik. Celem jest nauczenie modelu przewidywania wyników na podstawie nowych, nieoznakowanych danych.

Przykładem może być problem klasyfikacji zdjęć kotów i psów. Model jest trenowany na zbiorze zdjęć, które są odpowiednio oznaczone jako "kot" lub "pies".
Na podstawie tych danych model uczy się rozpoznawać wzorce i cechy charakterystyczne dla każdej kategorii, a następnie jest w stanie klasyfikować nowe zdjęcia.

Uczenie nadzorowane dzieli się na dwa główne typy problemów:
- Regresja: Model przewiduje wartość ciągłą, np. przewidywanie cen nieruchomości na podstawie danych historycznych.

- Klasyfikacja: Model przewiduje przynależność do jednej z predefiniowanych klas, np. rozpoznawanie spamu w wiadomościach e-mail.
Uczenie nienadzorowane (Unsupervised Learning) W przypadku uczenia nienadzorowanego model trenuje się na danych, które nie mają etykiet.

Celem jest odkrycie ukrytych struktur, wzorców lub związków w danych. Uczenie nienadzorowane jest szczególnie przydatne, gdy nie posiadamy oznaczonych danych lub gdy chcemy eksplorować dane w celu zrozumienia ich struktury.

Przykładem może być klastrowanie, gdzie model grupuje podobne obiekty w klastry.
Na przykład w analizie rynku model może podzielić klientów na różne segmenty na podstawie ich zachowań zakupowych, co pozwala firmom na lepsze dostosowanie swoich strategii marketingowych.

Uczenie przez wzmacnianie (Reinforcement Learning) Uczenie przez wzmacnianie różni się od powyższych metod, ponieważ model uczy się poprzez interakcję z otoczeniem.
Model podejmuje działania i na podstawie wyników tych działań otrzymuje nagrody lub kary. Celem jest maksymalizacja sumy nagród w długim okresie.

Uczenie przez wzmacnianie jest szeroko stosowane w grach komputerowych, robotyce i autonomicznych systemach. Przykładem może być system AI, który uczy się grać w grę wideo, gdzie model optymalizuje swoje działania w celu uzyskania jak najwyższego wyniku.

2.3 Przykłady algorytmów uczenia maszynowego Uczenie maszynowe obejmuje szeroki zakres algorytmów, z których każdy ma swoje specyficzne zastosowania i zalety. Oto kilka popularnych algorytmów:

Regresja Liniowa (Linear Regression) Regresja liniowa to jeden z najprostszych algorytmów uczenia nadzorowanego, stosowany do przewidywania wartości ciągłych.
Model regresji liniowej próbuje dopasować prostą linię do danych, minimalizując różnicę między przewidywanymi a rzeczywistymi wartościami.
Na przykład regresja liniowa może być używana do przewidywania cen domów na podstawie takich cech, jak powierzchnia, liczba pokoi czy lokalizacja.

Drzewa Decyzyjne (Decision Trees) Drzewa decyzyjne są wszechstronnym narzędziem do klasyfikacji i regresji. Model składa się z serii węzłów, które reprezentują pytania dotyczące danych, a odpowiedzi prowadzą do kolejnych węzłów, aż do uzyskania końcowej decyzji lub przewidywania.

Drzewa decyzyjne są łatwe do interpretacji i wizualizacji, co czyni je popularnym wyborem w wielu zastosowaniach. K-Nearest Neighbors (K-NN) Algorytm K-NN jest prostym, ale skutecznym narzędziem do klasyfikacji i regresji.

Polega on na identyfikacji k najbliższych sąsiadów (na podstawie określonej metryki odległości) dla danego punktu danych, a następnie przewidywaniu etykiety lub wartości na podstawie głosów większości lub średniej tych sąsiadów. K-NN jest stosowany w rozpoznawaniu obrazów, klasyfikacji tekstów oraz rekomendacjach produktów. K-Means i klastrowanie K-Means to popularny algorytm klastrowania, stosowany w uczeniu nienadzorowanym.
Algorytm grupuje dane w k klastrów, tak aby punkty w tym samym klastrze były do siebie jak najbardziej podobne, a punkty w różnych klastrach były jak najbardziej różne.

K-Means jest często używany w segmentacji klientów, analityce rynku i eksploracji danych.

Support Vector Machines (SVM) SVM to potężny algorytm klasyfikacyjny, który działa poprzez znalezienie hiperpowierzchni maksymalizującej margines między klasami w danych.

SVM jest szczególnie skuteczny w przypadku danych o wysokiej wymiarowości i jest stosowany w rozpoznawaniu obrazów, bioinformatyce i tekstowym przetwarzaniu języka naturalnego.

2.4 Proces tworzenia modelu ML Proces tworzenia modelu uczenia maszynowego składa się z kilku kluczowych etapów, które mają na celu zapewnienie, że model będzie skutecznie uczył się na danych i dostarczał dokładnych przewidywań.

Oto kroki, które zwykle obejmuje proces tworzenia modelu ML:
Zbieranie danych Dane są fundamentem każdego projektu ML.
Bez odpowiednich danych, nawet najlepszy algorytm nie będzie w stanie działać skutecznie.
Proces zbierania danych może obejmować różne źródła, takie jak bazy danych, czujniki, aplikacje webowe czy publiczne zbiory danych.

Kluczowe jest, aby dane były reprezentatywne dla problemu, który model ma rozwiązać.
Przygotowanie danych Dane rzadko są idealne od razu po zebraniu.
Często zawierają braki, błędy lub mogą być zapisane w formacie, który nie jest odpowiedni do analizy. Proces przygotowania danych obejmuje:

- Czyszczenie danych (usuwanie brakujących lub błędnych wartości).
- Normalizację i standaryzację danych (przekształcanie danych do zakresu odpowiedniego dla algorytmu).
- Tworzenie nowych cech (feature engineering) na podstawie dostępnych danych, aby zwiększyć skuteczność modelu.

Trenowanie modelu Trenowanie to proces, w którym model uczy się na podstawie danych treningowych.

Algorytm analizuje dane wejściowe i dostosowuje swoje parametry, aby jak najlepiej dopasować przewidywania do rzeczywistych wynikówz danych.

Podczas treningu model iteracyjnie przetwarza dane, ucząc się wzorców, które pozwolą mu na dokładne przewidywanie wyników dla nowych danych.

Ważnym aspektem jest unikanie tzw. przetrenowania (overfitting), czyli sytuacji, w której model staje się zbyt dopasowany do danych treningowych, co powoduje spadek jego ogólnej skuteczności na danych testowych. Walidacja modelu Walidacja modelu to etap, w którym sprawdzamy, jak dobrze model radzi sobie z danymi, których wcześniej nie widział.

Zazwyczaj dane są dzielone na trzy zestawy:

- Zbiór treningowy: używany do trenowania modelu.
- Zbiór walidacyjny: używany do dostrajania parametrów modelu i wyboru najlepszego algorytmu.
-m Zbiór testowy: używany do ostatecznej oceny modelu.

Jedną z popularnych technik walidacyjnych jest k-krotna walidacja krzyżowa (k-fold crossvalidation), która polega na podziale danych na k równych części.
Model jest trenowany k razy, za każdym razem używając innej części danych jako zbioru walidacyjnego, a pozostałych k-1 części jako zbioru treningowego.

Taka procedura pozwala na lepszą ocenę ogólnej skuteczności modelu. Optymalizacja hiperparametrów Hiperparametry to parametry modelu, które muszą być ustawione przed procesem treningu i nie są optymalizowane w trakcie samego trenowania (np. liczba drzew w lesie losowym, wartość k w algorytmie K-NN).

Proces optymalizacji hiperparametrów polega na eksperymentowaniu z różnymi ich wartościami, aby znaleźć te, które maksymalizują wydajność modelu na zbiorze walidacyjnym.

Techniki optymalizacji hiperparametrów obejmują:
- Przeszukiwanie siatki (grid search): przeszukiwanie wszystkich możliwych kombinacji hiperparametrów w ustalonym zakresie.
- Przeszukiwanie losowe (random search): losowe próbkowanie przestrzeni hiperparametrów.

Ewaluacja modelu Po zakończeniu treningu i optymalizacji hiperparametrów, model jest oceniany na zbiorze testowym, który nie był używany podczas treningu ani walidacji.

Ewaluacja modelu ma na celu ocenę jego ogólnej zdolności do przewidywania na nowych, nieznanych danych.

W zależności od problemu można używać różnych metryk do oceny modelu, takich jak:

- Dokładność (accuracy): odsetek poprawnie sklasyfikowanych przypadków.

- Precyzja (precision) i czułość (recall):
stosowane w przypadku nierównowagi klas w problemach klasyfikacji.

- Średni błąd bezwzględny (MAE) lub średni błąd kwadratowy (MSE) w przypadku regresji.

Implementacja modelu Jeśli model działa zgodnie z oczekiwaniami, można go zaimplementować w systemie produkcyjnym.
Implementacja obejmuje wdrożenie modelu w środowisku, gdzie będzie przetwarzać rzeczywiste dane i dostarczać przewidywania lub decyzje w czasie rzeczywistym.

Ważne jest, aby monitorować wydajność modelu po wdrożeniu, ponieważ zmieniające się dane mogą wpłynąć na jego skuteczność.
Monitorowanie i utrzymanie Model uczenia maszynowego nie kończy swojego cyklu życia na wdrożeniu.

Ważne jest ciągłe monitorowanie jego działania, aby upewnić się, że nadal dostarcza dokładnych wyników.
Może być konieczne ponowne trenowanie modelu, jeśli dane wejściowe zmienią się w czasie lub jeśli model zacznie tracić na skuteczności.
Proces ten nazywa się ciągłym uczeniem (continuous learning).