

Nauczanie maszynowe

Wprowadzenie

Koncepcja uczenia maszynowego została po raz pierwszy wprowadzona w latach 50. XX wieku, które były niezwykle w czasach pionierów AI. W 1950 roku Alan Turing opublikował artykuł „Computing Machinery and Intelligence”, w którym zasugerowano słynny test Aevaluation, który znamy dzisiaj jako Test Turinga. W 1959 r. Arthur Lee Samuel ukuł termin „uczenie maszynowe”. Uczenie maszynowe (ML) można szeroko zdefiniować jako metody obliczeniowe wykorzystujące doświadczenie w celu poprawy wydajności lub dokładnych prognoz. Definiujemy uczenie maszynowe jako serię matematycznych manipulacji wykonywanych na ważnych danych w celu uzyskania cennych informacji. Jest to badanie algorytmów, które uczą się na przykładach i doświadczeniu zamiast na sztywnych zasadach. Zwykle istnieją trzy główne typy problemów z uczeniem maszynowym: nadzorowane, nienadzorowane i wzmacniające.

* Problemy z nadzorowanym uczeniem maszynowym to problemy, w których chcemy dokonywać prognoz na podstawie zestawu przykładów.

* Problemy związane z nienadzorowanym uczeniem maszynowym to problemy, w których nasze dane nie mają zestawu zdefiniowanych zestaw kategorii, ale zamiast tego szukamy algorytmów uczenia maszynowego, które pomogą nam uporządkować dane.

Oznacza to, że nadzorowane problemy z uczeniem maszynowym zawierają zestaw danych historycznych, które chcemy wykorzystać do przewidywania przyszłości, problemy z nadzorowanym uczeniem maszynowym mają zestaw danych, których szukamy w uczeniu maszynowym, aby pomóc nam zorganizować lub zrozumieć.

* Wzmocnienie obejmuje określone zadanie lub cel, który musi wykonać system. W trakcie całego procesu otrzymuje informacje zwrotne w celu poznania pożądanych zachowań. Na przykład system napotyka błąd podczas wykonywania akcji lub nagrodę za osiągnięcie najbardziej korzystnego wyniku. W ten sposób program jest w stanie nauczyć się najbardziej skutecznego podejścia za pomocą sygnałów wzmacniających.

Wydaje się, że eksploracja danych i odkrywanie wiedzy w bazach danych (KDD) rozwiązują wyłącznie główny problem nauki o danych, ale uczenie maszynowe zwiększa efektywność biznesową. Techniki ML można z grubsza podzielić na cztery odrębne obszary: klasyfikacja, grupowanie, uczenie się skojarzeń i predykcja numeryczna. Klasyfikacja zastosowana do tekstu jest przedmiotem kategoryzacji tekstu, która polega na automatycznym sortowaniu zestawu dokumentów na kategorie (lub klasy lub tematy) ze wstępnie zdefiniowanego zestawu. Bezpośrednia klasyfikacja dokumentów jest stosowana przy indeksowaniu dokumentów w systemach wyszukiwania informacji, filtrowaniu tekstu (w tym ochronie przed spamem e-mail), kategoryzacji stron internetowych i wielu innych aplikacjach. Klasyfikacji można także używać do mniejszych części tekstu (akapitów, zdań, słów), w zależności od konkretnego zastosowania, np. segmentacja dokumentów, śledzenie tematów lub rozróżnianie słów. W podejściu opartym na uczeniu maszynowym algorytmy klasyfikacji (klasyfikatory) są wcześniej szkolone na temat uprzednio posortowanych etykietowanych danych, zanim zostaną zastosowane do sortowania niewidocznych tekstów. Zastosowanie technik klastrowania z tekstem można osiągnąć na dwóch poziomach. Analiza zbiorów dokumentów poprzez identyfikację klastrów podobnych wymaga niewiele więcej niż wykorzystanie znanych algorytmów klastrowania w połączeniu z miernikami podobieństwa dokumentów. W ramach klastrowania dokumentów klastrowanie może być nieco trudniejsze, ponieważ wymaga wstępnego przetwarzania tekstu i izolowania obiektów do grupowania - zdań, słów lub konstrukcji wymagającej wprowadzenia.

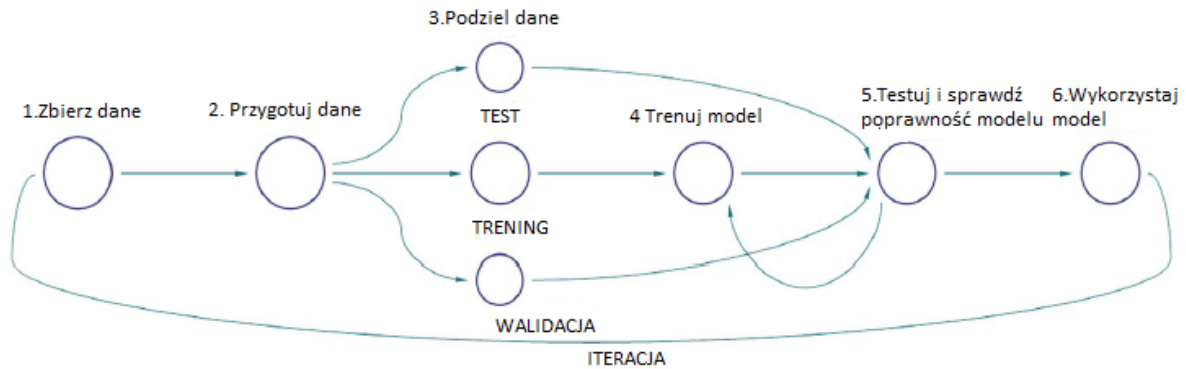
Uczenie się asocjacji jest zasadniczo uogólnieniem klasyfikacji, która ma na celu uchwycenie związków między dowolnymi cechami (zwanymi także atrybutami) przykładów w zbiorze danych. W tym sensie klasyfikacja przechwytyje tylko relacje wszystkich cech do jednej cechy określającej klasę. Proste zastosowanie uczenia asocjacyjnego do tekstu nie jest bardzo wykonalne ze względu na wysoką wymiarowość reprezentacji dokumentów, czyli znaczną liczbę cech (z których wiele może nie być bardzo pouczających).

Wykorzystywanie uczenia się przez stowarzyszenie na temat informacji uzyskanych z tekstu (na przykład za pomocą klasyfikacji i / lub grupowania) to inna historia i może dostarczyć wielu przydatnych informacji. Prognozowanie numeryczne (zwane również regresją, w szerszym znaczeniu tego słowa) jest kolejnym uogólnieniem klasyfikacji, w której cecha klasy nie jest dyskretna, lecz ciągła. Ta niewielka zmiana definicji powoduje ogromne różnice w wewnętrznych elementach algorytmów klasyfikacji i regresji. Jednak dzieląc przewidywaną cechę liczbową na skończoną liczbę przedziałów, do algorytmów można również zastosować każdy algorytm regresji. Przeciwnie, zwykle nie jest możliwe. Ponownie, podobnie jak w przypadku uczenia się asocjacyjnego, proste zastosowanie regresji w tekście nie jest szczególnie przydatne, z wyjątkiem klasyfikacji (szczególnie gdy wymagana jest miara przekonania, ale można to osiągnąć również za pomocą większości algorytmów klasyfikacji). Istnieje różnica między eksploracją danych a bardzo popularnym uczeniem maszynowym. Mimo to uczenie maszynowe polega na tworzeniu algorytmów w celu uzyskania cennych informacji, jest mocno skoncentrowane na ciągłym użytkowaniu w dynamicznie zmieniających się środowiskach i kładzie nacisk na korekty, przekwalifikowanie i aktualizację algorytmów w oparciu o wcześniejsze doświadczenia. Celem uczenia maszynowego jest stałe dostosowanie się do nowych danych i odkrywanie w nich nowe wzorce lub reguły. Czasami można to zrealizować bez ludzkiej pomocy i wyraźnego przeprogramowania. Uczenie maszynowe jest obecnie najdynamiczniej rozwijającą się dziedziną nauki o danych ze względu na szereg ostatnich przełomów teoretycznych i technologicznych. Doprowadziły do przetwarzania języka naturalnego, rozpoznawania obrazów, a nawet generowania nowych obrazów, muzyki i tekstów przez maszyny. Uczenie maszynowe pozostaje głównym instrumentem "budowania sztucznej inteligencji. Aby skorzystać z uczenia maszynowego w aplikacji lub nawet się go nauczyć, istnieją dwa sposoby. Po pierwsze, uczą się, jak korzystać z bibliotek, które działają jak czarna skrzynka, czyli zapewniają różne funkcje. Po drugie, aby nauczyć się pisać algorytmy i znaleźć współczynniki, dopasować model, znaleźć punkty optymalizacji i wiele więcej, abyś mógł wybrać swoją aplikację zgodnie z wymaganiami. Jeśli jednak chcesz się pobawić, istnieje kilka bibliotek i interfejsów programowania aplikacji, które pomogą Ci wykonać zadanie. Firmy używają technologii uczenia maszynowego do analizy historii zakupów swoich klientów i przygotowywania spersonalizowanych rekomendacji produktów na kolejny zakup. Ta zdolność do przechwytywania, analizowania i wykorzystywania danych klientów w celu spersonalizowania zakupów to przyszłość sprzedaży i marketingu. W sektorze transportu, w oparciu o historię podróży i sposób podróżowania różnymi trasami, uczenie maszynowe może pomóc firmom transportowym przewidzieć potencjalne problemy, które mogą wystąpić na niektórych trasach, i odpowiednio doradzić klientom, aby wybrali inną trasę. Firmy transportowe i logistyczne stopniowo wykorzystują technologię uczenia maszynowego do przeprowadzania analizy danych i modelowania danych, aby podejmować świadome decyzje i pomagać klientom w podejmowaniu trafnych decyzji podczas podróży.

Proces uczenia maszynowego

Główną różnicą między uczeniem maszynowym a tradycyjnie programowanymi algorytmami jest zdolność przetwarzania danych bez wyraźnego programowania. Oznacza to, że inżynier nie musi dostarczać maszynie szczegółowych instrukcji dotyczących sposobu traktowania każdego typu rekordu danych. Zamiast tego maszyna sama określa te reguły, opierając się na danych wejściowych.

Niezależnie od aplikacji do uczenia maszynowego, podstawowy przepływ pracy pozostaje taki sam i iteracyjnie się powtarza, gdy wyniki stają się datowane lub wymagają wyższej dokładności. Ta sekcja koncentruje się na wprowadzeniu podstawowych pojęć, które składają się na proces uczenia maszynowego, jak pokazano poniżej



Przepływ pracy wykonuje następujące kroki:

1. Zbierz dane. Skorzystaj z infrastruktury IT, aby zebrać jak najwięcej odpowiednich rekordów i połączyć je w zestaw danych.
2. Przygotuj dane. Przygotuj swoje dane do przetwarzania w najlepszy praktyczny sposób. Procedury przetwarzania i czyszczenia danych mogą być dość skomplikowane, ale ogólnie mają na celu uzupełnienie brakujących wartości i skorygowanie innych wad danych, takich jak różne przedstawienia tych samych wartości w kolumnie.
3. Podziel dane. Oddziel podzbiory danych, aby wyszkolić model i dalej oceniać jego skuteczność względem nowych danych.
4. Trenuj model. Użyj podzbioru danych historycznych, aby umożliwić algorytmowi rozpoznanie zawartych w nim wzorców.
5. Przetestuj i sprawdź poprawność modelu. Oceń wydajność modelu przy użyciu podzbiorów testowania i weryfikacji danych historycznych i zrozum, jak dokładne jest przewidywanie.
6. Wykorzystaj model. Osadź przetestowany model w kontekście decyzyjnym jako część rozwiązania analitycznego.
7. iteracja. Zbierz nowe dane po użyciu modelu, aby stopniowo go ulepszać.

Podstawowym artefaktem każdego wykonania uczenia maszynowego jest model matematyczny, który opisuje, w jaki sposób algorytm przetwarza nowe dane po przeszkoleniu z podzbiorem danych historycznych. Celem szkolenia jest opracowanie modelu zdolnego do sformułowania wartości docelowej (atrybutu), jakiejś nieznannej wartości każdego obiektu danych. Na przykład musisz przewidzieć, czy klienci Twojego sklepu e-commerce dokonają zakupu, czy odeszli. Te prognozy kupują lub opuszczają cele atrybutu, których szukamy. Aby wytrenować model w zakresie wykonywania tego typu prognoz, „karmisz” algorytm zestawem danych, który przechowuje różne zapisy zachowań klientów i wyników, na przykład, czy klienci odeszli, czy sfinalizowali zakup. Na podstawie tych danych historycznych model będzie mógł przewidywać przyszłe dane.

Algorytmy uczenia się

Regresja liniowa

Model liniowy wykorzystuje prostą formułę do znalezienia linii „najlepiej dopasowanej” poprzez zestaw punktów danych. Znaleźć zmienną, którą chcesz przewidzieć, poprzez równanie zmiennych, które znasz. Aby znaleźć prognozę, wprowadzamy zmienne, które znamy, aby uzyskać naszą odpowiedź. Innymi słowy, aby dowiedzieć się, ile czasu zajmie upieczenie ciasta, po prostu wprowadzamy składniki. Istnieją różne formy algorytmów modelu liniowego. Regresja liniowa, znana również jako „regresja najmniejszych kwadratów”, jest najbardziej standardową formą modelu liniowego. W przypadku problemów z regresją (zmienną, którą próbujemy przewidzieć jest liczbowa), regresja liniowa jest najprostszym modelem liniowym.

Algorytmy regresji są powszechnie stosowane do analizy statystycznej i są kluczowymi algorytmami do wykorzystania w uczeniu maszynowym. Algorytmy regresji pomagają analitykom modelować relacje między punktami danych. Algorytmy regresji definiują numeryczne wartości docelowe zamiast klas. Szacując zmienne numeryczne, algorytmy te są skuteczne w przewidywaniu popytu na produkty, wielkości sprzedaży, zwrotów marketingowych itp. Na przykład:

* Ile elementów tego produktu będziemy mogli sprzedać w tym roku?

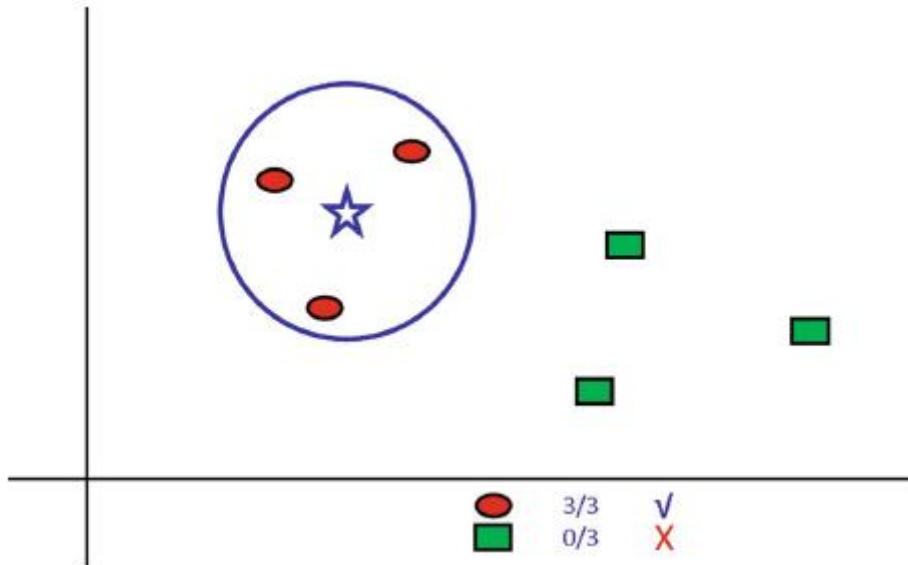
* Jaki będzie koszt podróży dla tego miasta?

* Jaka jest maksymalna prędkość, aby samochód mógł przetrwać?

Algorytmy regresji mogą kwantyfikować siłę korelacji między zmiennymi w zestawie danych. Ponadto analiza regresji może być przydatna do przewidywania przyszłych wartości danych na podstawie wartości historycznych. Należy jednak pamiętać, że analiza regresji zakłada, że korelacja dotyczy związku przyczynowego. Bez zrozumienia kontekstu wokół danych analiza regresji może prowadzić do niedokładnych prognoz. Regresja logistyczna to po prostu dostosowanie regresji liniowej do problemów klasyfikacyjnych (zmienną, którą próbujemy przewidzieć, jest odpowiedź „tak / nie”). Regresja logistyczna jest bardzo dobra w przypadku problemów z klasyfikacją ze względu na jej kształt. Zarówno regresja liniowa, jak i regresja logistyczna mają te same wady. Oba mają tendencję do „nadmiernego dopasowania”, co oznacza, że model dostosowuje się zbyt dokładnie do danych kosztem możliwości uogólnienia na dotychczas niewidoczne dane. Dlatego oba modele są często „regularyzowane”, co oznacza, że mają pewne kary, aby zapobiec nadmiernemu zużyciu. Kolejną wadą modeli liniowych jest to, że ponieważ są one proste, mają trudności z przewidywaniem bardziej złożonych zachowań.

k-Najbliższy sąsiad

Algorytm k-najbliższego sąsiada to metoda klasyfikacji obiektów na podstawie najbliższych przykładów szkoleniowych w przestrzeni cech. Sprawdza przestrzeń funkcji i może z pewnością dać prognozę na podstawie najbliższego sąsiada. Działa z tym, że obiekt, który jest w pobliżu, miałby podobne wartości predykcyjne, a gdy znamy wartość predykcyjną obiektu, łatwo przewidzieć dla najbliższego sąsiada. Algorytm k-najbliższego sąsiada jest jednym ze skromnych znanych algorytmów uczenia maszynowego i jest często nazywany leniwym uczniem, ponieważ zależy od przewidywań tylko z określonego wyboru instancji najbardziej podobnych do instancji zestawu testowego. Próbkę szkoleniową są opisane przez n-wymiarowe atrybuty numeryczne. Każda próbka reprezentuje punkt w przestrzeni n-wymiarowej; wszystkie próbki treningowe są przechowywane w n-wymiarowej przestrzeni wzorów. Gdy mamy nieznaną próbkę, algorytm przeszukuje przestrzeń wzorcową dla k próbek treningowych, które są najbliższej nieznannej próbki, k próbek treningowych to k „najbliższych sąsiadów” nieznannej próbki, jak pokazano poniżej



k-najbliższy sąsiad okazuje się być bardzo skuteczną metodą na hałaśliwe dane treningowe, k-najbliższy sąsiad dobrze sprawdza się w zakresie automatyzacji, ponieważ wiele algorytmów jest solidnych i daje dobrą prognozę, gdy mamy zestaw danych z brakującymi danymi. k-najbliższy sąsiad może zostać ulepszony, jeśli mamy wersję drzewa decyzyjnego do wstępnego przetwarzania zestawu danych. Przetwarzanie wstępne można zastosować na nasz zbiór danych na kilka sposobów, biorąc pod uwagę jego charakter; jeśli zestaw danych składa się z atrybutów numerycznych, algorytmy dyskretyzacji byłyby bardzo skuteczne, zmniejszając liczbę krotek zestawu danych do znacznie mniejszej liczby interwałów. Dane wyjściowe po dyskretyzacji byłyby podawane do najbliższego sąsiada k. Dzięki temu proces byłby znacznie szybszy, ponieważ rozważa się mniej krotek.

Drzewa decyzyjne

Drzewo decyzyjne działa w formie struktury drzewa od góry, zwanej węzłem głównym aż do liści; każda z gałęzi reprezentuje wynik testu, a węzły liści reprezentują klasy. Aby sklasyfikować dowolną nieznaną próbkę, testujemy atrybut próbki w stosunku do drzewa decyzyjnego. Ścieżka jest śledzona od korzenia, czyli od szczytu drzewa do węzła liścia, który zawiera predykcję klasy dla tej próbki. Drzewo decyzyjne jest podatne na dużo hałasu, a standardową techniką radzenia sobie z tym jest przycinanie drzewa. Przycinanie polega na usunięciu dowolnego stanu w jego poprzedniku, który nie poprawia szacunkowej dokładności reguły. Ten proces ma na celu poprawę dokładności klasyfikacji dla niewidocznych danych. Aby utworzyć lub wyszkolić drzewo decyzyjne, bierzemy dane, których użyliśmy do wyszkolenia modelu i znajdujemy, które atrybuty najlepiej dzielą zestaw pociągów w odniesieniu do celu. Na przykład drzewo decyzyjne można wykorzystać w przypadku oszustwa związanego z kartą kredytową. Atrybutem, który najlepiej przewiduje ryzyko oszustwa, jest kwota zakupu (na przykład osoba posiadająca kartę kredytową dokonała bardzo dużego zakupu). Może to być pierwszy podział (lub rozgałęzienie) - te karty, które mają wyjątkowo wysokie zakupy, i te, które tego nie robią. Następnie używamy drugiego najlepszego atrybutu (np. często używana jest karta kredytowa), aby utworzyć następny podział. Następnie możemy kontynuować, dopóki nie będziemy mieć wystarczających atrybutów, aby zaspokoić nasze potrzeby. Algorytmy klasyfikacji określają, do której kategorii należą obiekty ze zbioru danych. Dlatego kategorie są zwykle powiązane z klasami. Rozwiązując problemy z klasyfikacją, możesz odpowiedzieć na różne pytania:

* Czy ten e-mail jest spamem, czy nie?

* Czy ta transakcja jest podwójna, czy nie?

* Jaki rodzaj produktu kupujący chętniej kupuje: sofę, stół jadalny lub krzesła ogrodowe?

Skalowalność jest istotnym problemem w drzewie decyzyjnym, ponieważ nie skaluje się dobrze na dużych zestawach danych, a podczas eksploracji danych typowe zestawy szkoleniowe obejmują miliony próbek. Problem skalowalności powstaje, ponieważ zestawy szkoleniowe są przechowywane w pamięci głównej. Ograniczenie to ogranicza skalowalność takich algorytmów, w których konstrukcja drzewa decyzyjnego może stać się nieefektywna z powodu zamiany próbek szkoleniowych do iz pamięci głównej i pamięci podręcznej. Opcją jest dyskretyzacja ciągłych atrybutów i przeprowadzanie próbkowania w każdym węźle. Ale ma to również swoją nieefektywność. Inną opcją jest podzielenie dużego drzewa decyzyjnego na podzbiory i zbudowanie drzewa decyzyjnego z podzestawów. Ponieważ pracujemy tylko nad podzbiorami, dokładność naszego wyniku nie jest tak dobra, jakbyśmy wykorzystali cały zestaw danych. Cofnięty do drzewa decyzyjnego jest chciwość algorytmu. Chciwość oznacza, że algorytm zbyt wcześnie podejmuje pewne wybory, co uniemożliwia później znalezienie najlepszego ogólnego rozwiązania. Drzewa decyzyjne są bardzo szybkie, a dokładność klasyfikacji jest naturalnie wysoka dla danych, w których mapowanie klas składa się z długich i cienkich obszarów w przestrzeni koncepcji. Ulepszeniem tej techniki uczenia się może być modyfikacja algorytmu w celu obsługi atrybutów o stałej wartości. Drzewo decyzyjne ma cechę odporności na wiele typów predyktorów. To sprawia, że dobrze nadaje się jako dobra metoda wstępnego przetwarzania dla innych algorytmów. Przykładem jest wstępne przetwarzanie danych dla sieci neuronowych; ze względu na swoją szybkość dogodnie wykonałby pierwszy przekaz danych, które utworzyłyby podzbiór predyktorów, które zostałyby wprowadzone do sieci neuronowej lub najbliższego sąsiada. To zdecydowanie zmniejszyłoby poziom szumów, z którym musi radzić sobie sieć neuronowa, a to zdecydowanie poprawiłoby wydajność sieci neuronowej. Kolejnym bardzo precyzyjnym zadaniem klasyfikacji jest wykrywanie anomalii. Zazwyczaj jest uznawany za klasyfikację jednej klasy, ponieważ celem wykrywania anomalii jest znalezienie wartości odstających, nietypowych obiektów w danych, które nie pojawiają się w normalnym rozkładzie. Jakie problemy może rozwiązać:

* Czy w naszym zbiorze danych są klienci o wyróżniających się cechach?

* Czy zauważamy nietypowe zachowania wśród naszych klientów ubezpieczeniowych?

Konstrukcja funkcji i redukcja danych

Rola reprezentacji została uznana za kluczową kwestię w AI i ML. W paradygmacie uczenia się na przykładach i reprezentacji wartości atrybutów danych wejściowych oryginalna reprezentacja jest wektorem atrybutów (cech, zmiennych) opisujących przykłady (instancje, obiekty). Proces transformacji atrybutów wejściowych, stosowanych w konstrukcji elementów, można sformułować w następujący sposób: biorąc pod uwagę oryginalny wektor cech i zestaw szkoleniowy, stwórz pochodną reprezentację, która jest lepsza, biorąc pod uwagę niektóre kryteria (tj. dokładność predykcyjna, wielkość reprezentacji). Nowe przekształcone atrybuty albo zastępują oryginalne atrybuty, albo można je dodać do opisu przykładów. Przykłady transformacji atrybutów to zliczanie, grupowanie, konstrukcja / dyskretyzacja przedziałów, skalowanie, spłaszczanie, normalizacja (wartości liczbowych), grupowanie, analiza głównych składowych itp. Wiele transformacji jest możliwych dzięki zastosowaniu wszelkiego rodzaju wzorów matematycznych, ale w praktyce tylko ograniczona liczba transformacji jest skuteczna.

Losowy las

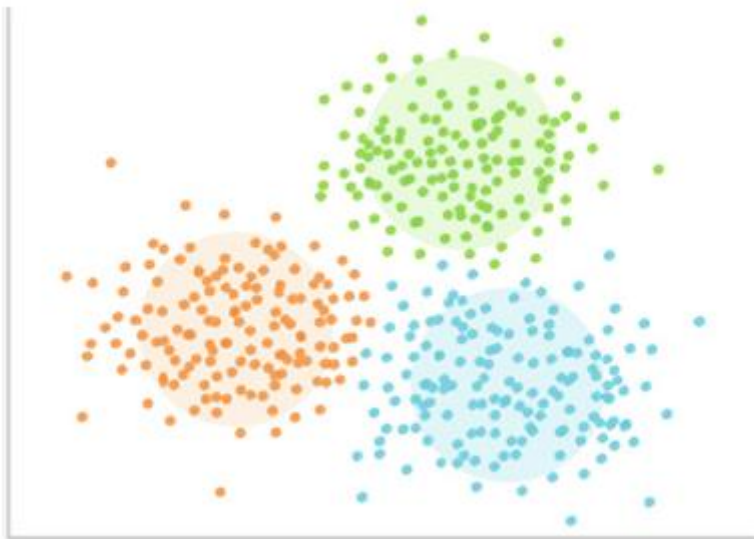
Losowy las to średnia z kilku drzew decyzyjnych, z których każde jest trenowane z losową próbką danych. Każde pojedyncze drzewo w lesie jest słabsze niż pełne drzewo decyzyjne, ale łącząc je wszystkie, uzyskujemy lepszą ogólną wydajność dzięki różnorodności. Losowy las jest dziś bardzo rozpowszechnionym algorytmem uczenia maszynowego. Jest bardzo łatwy do trenowania i ma dobre wyniki. Jego wadą jest to, że generowanie prognoz w stosunku do innych algorytmów może być powolne, więc możesz nie używać go, gdy potrzebujesz błyskawicznych prognoz. Losowy las daje znacznie dokładniejsze prognozy w porównaniu do modeli regresji w wielu scenariuszach. Przypadki te mają na ogół dużą liczbę zmiennych predykcyjnych i ogromną wielkość próby. Wynika to z tego, że rejestruje wariację kilku zmiennych wejściowych jednocześnie i umożliwia udział dużej liczby obserwacji w przewidywaniu.

Algorytm k-średnich

k-średnich jest rodzajem nienadzorowanego algorytmu, który rozwiązuje problem klastrowania. Główną różnicą między regularną klasyfikacją a klastrowaniem jest to, że algorytm ma wyzwanie grupowania elementów w klastrach bez predefiniowanych klas. Oznacza to, że powinien on decydować o zasadach samego podziału bez ludzkiego kierownictwa. Analiza skupień jest zwykle realizowana w ramach nienadzorowanego stylu uczenia się. Klastrowanie może rozwiązać następujące problemy:

- * Jakie są główne segmenty klientów, biorąc pod uwagę ich dane demograficzne i zachowania?
- * Czy istnieje związek między ryzykiem niewykonania zobowiązań przez niektórych klientów banków a ich zachowaniami?
- * Jak możemy sklasyfikować słowa kluczowe, których używają ludzie, aby wejść na naszą stronę internetową?

Jego procedura przebiega w prosty i bezpośredni sposób, aby sklasyfikować dany zestaw danych przez pewną liczbę klastrów (zakładając k klastrów), patrz ryc. 3. Punkty danych wewnątrz klastra są jednorodne i niejednorodne względem grup rówieśniczych



Jak k-średnia tworzy klaster:

1. k -średnia wybiera liczbę k punktów dla każdej gromady znanej jako centroidy.
2. Każdy punkt danych tworzy klaster z najbliższymi centroidami, czyli k klastrami.
3. Znajduje środek ciężkości każdego klastra w oparciu o istniejące elementy klastra. Tutaj mamy nowe centroidy.
4. Ponieważ mamy nowe centroidy, powtórz krok 2 i 3. Znajdź najbliższą odległość dla każdego punktu danych od nowych centroidów i skojarz się z nowymi k -klastrami. Powtarzaj ten proces, aż dojdzie do konwergencji, to znaczy centroidy się nie zmienią.

W k -średnich mamy klastry, a każdy klaster ma własny centroid. Suma kwadratów różnicy między środkiem ciężkości a punktami danych w klastrze stanowi sumę wartości kwadratowej dla tego klastra. Po dodaniu sum wartości kwadratowych dla wszystkich klastrów staje się ona sumą w ramach sumy wartości kwadratowej dla rozwiązania klastrowego. Wiemy, że wraz ze wzrostem liczby skupień wartość ta stale maleje, ale jeśli narysujesz wynik, możesz zauważyć, że suma do kwadratu odległości gwałtownie spada do pewnej wartości k , a następnie znacznie wolniej. Tutaj możemy znaleźć optymalną liczbę klastrów.

Redukcja wymiarów

Redukcja wymiarów pomaga systemom usuwać dane, które nie są przydatne do analizy. Ta grupa algorytmów służy do usuwania zbędnych danych, wartości odstających i innych nieprzydatnych danych. Redukcja wymiarów może być pomocna podczas analizy danych z czujników i innych przypadków użycia Internetu rzeczy (IoT). W systemach IoT mogą istnieć tysiące punktów danych po prostu informujących, że czujnik jest włączony. Przechowywanie i analiza, że „włączone” dane nie są pomocne i zajmą ważne miejsce do przechowywania. Ponadto, usuwając te zbędne dane, poprawi się wydajność systemu uczenia maszynowego. Wreszcie zmniejszenie wymiarów pomoże również analitykom wizualizować dane.

Nauka przez wzmocnienie

Uczenie się przez wzmocnianie to komputerowe podejście do zrozumienia i automatyzacji uczenia się zorientowanego na cel i podejmowania decyzji. Uczy się, co robić i jak mapować sytuacje do działań. Wynikiem jest maksymalizacja liczbowego sygnału nagrody. Uczeń nie jest informowany o tym, jaką akcję podjąć, ale musi odkryć, która akcja da maksymalną nagrodę. Uczenie się przez wzmocnienie jest definiowane nie przez charakteryzowanie algorytmów uczenia się, ale przez charakteryzowanie problemu uczenia się. Każdy algorytm, który dobrze nadaje się do rozwiązania tego problemu, uważamy za algorytm uczenia wzmocniającego. Uczenie się ze wzmocnieniem różni się od nauki nadzorowanej, czyli rodzaju nauki, którą badamy. Nadzorowane uczenie się polega na uczeniu się na przykładach dostarczonych przez pewnego doświadczonego zewnętrznego opiekuna. Jest to ważny rodzaj uczenia się, ale sam w sobie nie jest odpowiedni do uczenia się na podstawie interakcji. W przypadku problemów interaktywnych często niepraktyczne jest uzyskanie przykładów pożądaných zachowań, które są zarówno prawidłowe, jak i reprezentatywne dla wszystkich sytuacji, w których agent musi działać. Na nieznanym terytorium - gdzie można oczekiwać, że nauka będzie najbardziej korzystna - agent musi być w stanie uczyć się na podstawie własnego doświadczenia. Jednym z wyzwania, które pojawiają się we wzmocnianiu uczenia się, a nie w innych rodzajach uczenia się, jest kompromis między eksploracją a eksploatacją. Aby uzyskać dużą nagrodę, agent uczący się od wzmocniania musi preferować działania, które próbował w przeszłości i które okazały się skuteczne w tworzeniu nagrody. Ale aby odkryć takie akcje, musi spróbować akcji, których wcześniej nie wybrał. Agent musi wykorzystać to, co już wie, aby uzyskać nagrodę, ale musi także zbadać, aby w przyszłości

dokonywać lepszych wyborów akcji. Dylemat polega na tym, że ani eksploatacja, ani eksploracja nie mogą być realizowane wyłącznie bez niepowodzenia zadania. Agent musi wypróbować różne działania i stopniowo faworyzować te, które wydają się najlepsze. Przy stochastycznym zadaniu każdą akcją trzeba wielokrotnie wypróbować, aby wiarygodnie oszacować oczekiwaną nagrodę. Matematycy intensywnie badają dylemat poszukiwawczo-wydobywczy od wielu dziesięcioleci. Inną kluczową cechą uczenia się przez wzmocnienie jest to, że wyraźnie bierze on pod uwagę cały problem ukierunkowanego na cel agenta wchodzącego w interakcję z niepewnym środowiskiem. Jest to w przeciwieństwie do wielu podejść, które rozwiązują podproblemy, nie zajmując się tym, jak mogą się one zmieścić w większym obrazie. Wspomnieliśmy na przykład, że wiele badań dotyczących uczenia maszynowego dotyczy uczenia się nadzorowanego bez wyraźnego określenia, w jaki sposób taka umiejętność byłaby w końcu przydatna. Inni badacze opracowali teorie planowania z ogólnymi celami, ale bez uwzględnienia roli planowania w podejmowaniu decyzji w czasie rzeczywistym ani pytania, skąd wzięłyby się modele predykcyjne niezbędne do planowania. Chociaż podejścia te przyniosły wiele użytecznych wyników, ich skupienie się na izolowanych pod-problemach jest znaczącym ograniczeniem.

Uczenie się przez wzmocnienie ma odwrotny charakter, zaczynając od kompletnego, interaktywnego agenta dążącego do celu. Wszyscy agenci uczący się o wzmocnieniu mają wyraźne cele, potrafią wyczuć aspekty swojego środowiska i mogą wybrać działania, które wpłyną na ich środowisko. Co więcej, zwykle zakłada się od samego początku, że agent musi działać pomimo znacznej niepewności co do środowiska, z którym ma do czynienia. Gdy uczenie się przez wzmocnienie wymaga planowania, musi uwzględniać wzajemne oddziaływanie między planowaniem a wyborem działań w czasie rzeczywistym, a także pytanie, w jaki sposób pozyskiwane i ulepszone są modele środowiskowe. Kiedy uczenie się ze wzmocnieniem obejmuje uczenie się nadzorowane, robi to z bardzo konkretnych powodów, które określają, które zdolności są kluczowe, a które nie. Aby badania naukowe mogły się rozwijać, należy wyodrębnić i przeanalizować ważne podproblemy, ale powinny to być podproblemy motywowane jasnymi rolami w kompletnych, interaktywnych agentach poszukujących celu, nawet jeśli wszystkie szczegóły kompletnego agenta nie mogą być jeszcze wypełnione.

Zwiększanie gradientu

Zwiększanie gradientu, podobnie jak losowy las, powstaje również z „słabych” drzew decyzyjnych. Istotną różnicą jest to, że podczas zwiększania gradientu drzewa są trenowane jeden po drugim. Każde kolejne drzewo jest trenowane przede wszystkim z danymi, które zostały nieprawidłowo zidentyfikowane przez poprzednie drzewa. Pozwala to wzmocnieniu gradientu skupić się mniej na łatwych do przewidzenia przypadkach, a bardziej na trudnych przypadkach. Zespół jest tylko zbiorem predyktorów, który jest średnią wszystkich prognoz, aby dać ostateczną prognozę. Powodem, dla którego używamy zestawów jest to, że wiele różnych predyktorów próbujących przewidzieć tę samą zmienną docelową wykona lepszą pracę niż jakikolwiek pojedynczy predyktor sam. Techniki łączenia są dalej klasyfikowane do Bagging and Boosting. Wzmocnienie to technika zespolona, w której predyktory nie są tworzone niezależnie, ale sekwencyjnie. Tworzenie worków jest prostą techniką łączenia, w której budujemy wiele niezależnych predyktorów / modeli / uczniów i łączymy je przy użyciu niektórych technik uśredniania modeli. Zwiększanie gradientu jest przykładem algorytmu zwiększania. Trenuje się szybko i działa bardzo dobrze. Jednak niewielkie zmiany w zestawie danych treningowych mogą powodować radykalne zmiany w modelu, więc mogą nie dać najbardziej wytłumaczalnych wyników

Sieci neuronowe

Sieci neuronowe w biologii są wzajemnie połączonymi neuronami, które wymieniają między sobą wiadomości. Pomysł ten został teraz dostosowany do uczenia maszynowego i nazywa się go sztucznymi sieciami neuronowymi (ANN). Głębokie uczenie się, które jest najnowszą koncepcją, to tylko kilka warstw sztucznych sieci neuronowych umieszczonych jedna po drugiej. Sieć neuronowa to zestaw połączonych jednostek wejścia-wyjścia, w których każde połączenie ma przypisany ciężar. Sieć neuronowa składa się z formalnych neuronów, które są połączone w taki sposób, że każde wyjście neuronu służy ponadto jako wejście zasadniczo większej liczby neuronów podobnie, jak końce aksonów neuronu biologicznego są połączone poprzez wiązania synaptyczne z dendrytami innych neuronów. Liczba neuronów i sposób ich wzajemnego połączenia determinują architekturę (topologię) sieci neuronowej. Pod względem celu neurony wejściowe, pracujące (warstwa ukryta, mediacja) i wyjściowe można rozróżnić w sieci. Neurony wejściowe i wyjściowe reprezentują odpowiednio receptory i efekторы oraz tworzą połączone neurony robocze i odpowiednie kanały między nimi w celu propagacji odpowiednich sygnałów. Te kanały nazywane są ścieżkami w modelu matematycznym. Propagacja sygnału i przetwarzanie informacji wzdłuż ścieżki sieciowej realizowane są przez zmianę stanów neuronów na tej ścieżce. Stany wszystkich neuronów w sieci tworzą stan sieci neuronowej i związane z nią wagi synaptyczne i wszystkie połączenia reprezentują konfigurację sieci neuronowej. Sieć neuronowa rozwija się stopniowo w czasie, wzajemne połączenia, a także stany neuronu są zmieniane, a wagi dostosowywane. W kontekście aktualizacji tych atrybutów sieci w czasie przydatne jest podzielenie globalnej dynamiki sieci neuronowej na trzy dynamiki i rozważenie trzech faz działania sieci:

1. Architektoniczne (zmiana topologii): określa topologię sieci i jej możliwą zmianę. Aktualizacja architektury zwykle ma zastosowanie w ramach trybu adaptacyjnego w taki sposób, że sieć jest zaopatrywana w dodatkowe neurony i połączenia, gdy jest to potrzebne. Jednak w większości przypadków dynamika architektury zakłada stałą topologię sieci neuronowej, która już się nie zmienia.
2. Obliczeniowy (zmiana stanu): określa stan początkowy sieci i regułę jej aktualizacji w czasie, pod warunkiem, że topologia sieci i konfiguracja są ustalone. Na początku trybu obliczeniowego stany neuronów wejściowych są przypisywane do wejścia sieciowego, a pozostałe neurony znajdują się w stanie początkowym. Wszystkie potencjalne wejścia i stany sieciowe tworzą odpowiednio przestrzeń wejściową i stanowią sieci neuronowej. Po zainicjowaniu stanu sieci wykonywane jest właściwe obliczenie.
3. Adaptacyjny (zmiana konfiguracji): określa początkową konfigurację sieci i sposób, w jaki wagi w sieci są dostosowywane w czasie. Wszystkie potencjalne konfiguracje sieci tworzą przestrzeń wagową sieci neuronowej. Na początku trybu adaptacyjnego wagi wszystkich połączeń sieciowych są przypisywane do początkowej konfiguracji. Po zainicjowaniu konfiguracji sieci przeprowadzana jest odpowiednia adaptacja. Podobnie, jak w przypadku dynamiki obliczeniowej, ogólnie można wziąć pod uwagę model z ciągłą ewolucją wag sieci neuronowych, gdy konfiguracja jest ciągłą funkcją czasu zwykle opisywaną równaniem różniczkowym. Jednak w większości przypadków przyjmuje się dyskretny czas adaptacji.

Ta klasyfikacja nie odpowiada rzeczywistości neurofizjologicznej, ponieważ w układzie nerwowym wszystkie odpowiednie zmiany zachodzą jednocześnie. Powyżej wprowadzona dynamika sieci neuronowej jest zwykle określana przez warunek początkowy oraz równanie lub regułę matematyczną, która określa rozwój określonej cechy sieci (topologia, stan, konfiguracja) w czasie. Aktualizacje kontrolowane przez te reguły są wykonywane w odpowiednich trybach operacyjnych sieci neuronowej. Poprzez konkretyzację wprowadzonej dynamiki uzyskuje się różne modele sieci neuronowych, które są odpowiednie do rozwiązania określonych zadań. Oznacza to, że aby określić

konkretny model sieci neuronowej, wystarczy zdefiniować jego dynamikę architektoniczną, obliczeniową i adaptacyjną. Sieci neuronowe mają zdolność do klasyfikowania wzorców na danych, które nie zostały przeszkolone, a także mają lepszą tolerancję niż inny klasyfikator podczas obsługi hałaśliwych danych.

Słabością sieci neuronowych jest trudność w interpretacji jej wyników, które występują w formie symbolicznej. Inny klasyfikator jest lepszy pod tym względem. Sieć neuronowa pozwala na interakcje zmiennych wysokiego rzędu ze względu na zwiększoną łączność i ma tendencję do radzenia sobie ze skorelowanymi danymi lepiej niż inne techniki uczenia się. Sieci neuronowe mają szerokie zastosowanie do rzeczywistych problemów biznesowych. Faktem jest, że z powodzeniem zastosowano je w wielu branżach. Ponieważ sieci neuronowe najlepiej identyfikują wzorce lub trendy w danych, dobrze nadają się do prognozowania lub prognozowania, w tym:

- * Prognozowanie sprzedaży
- * Kontrola procesu przemysłowego
- * Badania klientów
- * Walidacji danych
- * Zarządzanie ryzykiem
- * Marketing docelowy

Istnieje wiele korzyści, które analityk osiąga dzięki wykorzystaniu sieci neuronowych w swojej pracy. Rozpoznawanie wzorców jest potężną techniką wykorzystywania informacji w danych i uogólniania ich. Sieci neuronowe uczą się rozpoznawać wzorce istniejące w zbiorze danych. System rozwija się poprzez uczenie się, a nie programowanie. Programowanie zajmuje znacznie więcej czasu analityka i wymaga od analityka określenia dokładnego zachowania modelu. Sieci neuronowe uczą się wzorców danych, uwalniając analityka do ciekawej pracy. Sieci neuronowe są elastyczne w zmieniającym się środowisku. Systemy oparte na regułach lub systemy zaprogramowane są ograniczone do sytuacji, dla której zostały zaprojektowane - gdy zmieniają się warunki, przestają obowiązywać. Chociaż sieci neuronowe mogą potrzebować trochę czasu, aby nauczyć się nagłej drastycznej zmiany, doskonale nadają się do adaptacji do ciągle zmieniających się informacji. Sieci neuronowe mogą budować modele informacyjne tam, gdzie zawodzą bardziej konwencjonalne podejścia. Ponieważ sieci neuronowe mogą obsługiwać bardzo złożone interakcje, mogą łatwo modelować dane, które są zbyt trudne do modelowania za pomocą tradycyjnych podejść, takich jak statystyki wnioskowania lub logika programowania. Wydajność sieci neuronowych jest co najmniej tak dobra jak klasyczne modelowanie statystyczne i lepsza w przypadku większości problemów. Sieci neuronowe budują modele, które lepiej odzwierciedlają strukturę danych w znacznie krótszym czasie. Sieci neuronowe działają teraz dobrze ze skromnym sprzętem komputerowym. Chociaż sieci neuronowe są intensywne obliczeniowo, procedury zostały zoptymalizowane do tego stopnia, że mogą teraz działać w rozsądnym czasie na komputerach osobistych. Nie wymagają superkomputerów, jak to miało miejsce we wczesnych dniach badań sieci neuronowej. Istnieją pewne ograniczenia dotyczące przetwarzania neuronowego. Kluczowym ograniczeniem jest niezdolność sieci neuronowej do wyjaśnienia użytego modelu. Analitycy często chcą wiedzieć, dlaczego model zachowuje się taki, jaki jest. Sieci neuronowe mają lepsze odpowiedzi, ale trudno im wyjaśnić, jak się tam dostały. Jest to powód, dla którego połączenie neuronowe zapewnia tak wiele narzędzi do eksploracji wyników i tak wiele opcji operacyjnych na narzędzia, które budują model. Eksperymentując z różnymi parametrami i w pełni badając wyniki zarówno w tekście, jak i grafice, użytkownicy połączeń neuronowych mogą lepiej zrozumieć

zachowanie modelu i większa pewność wyników. Jest kilka innych ograniczeń, które należy zrozumieć. Po pierwsze, trudno jest wyodrębnić reguły z sieci neuronowych. Czasami jest to ważne, że ludzie, którzy muszą wyjaśnić swoją odpowiedź innym i tym, którzy byli zaangażowani w sztuczną inteligencję, szczególnie systemy eksperckie oparte na regułach. W większości metod analitycznych nie można po prostu wyrzucić danych do sieci neuronowej i uzyskać rozstrzygającej odpowiedzi, ale trzeba poświęcić czas na zrozumienie problemu lub wyniku, który próbujesz przewidzieć. I musisz mieć pewność, że dane wykorzystane do szkolenia systemu są odpowiednie (tj. Odzwierciedlają zaangażowane czynniki) i są mierzone w sposób odzwierciedlający zachowanie tych czynników. Jeśli dane nie są reprezentatywne dla problemu, przetwarzanie neuronowe nie przyniesie dobrych wyników. Jest to klasyczna sytuacja, w której „wyrzucanie śmieci” z pewnością spowoduje „wyrzucanie śmieci”. Wreszcie, szkolenie modelu z bardzo złożonego zestawu danych może zająć trochę czasu. Techniki neuronowe są intensywnie obciążone komputerowo i będą powolne na komputerach klasy niskiej lub komputerach bez koprocesorów matematycznych. Należy jednak pamiętać, że całkowity czas na uzyskanie wyniku może być nadal szybszy niż w przypadku innych metod analizy danych, nawet jeśli system zajmuje więcej czasu. Sama szybkość przetwarzania nie jest jedynym czynnikiem wpływającym na wydajność, a sieci neuronowe nie wymagają programowania czasowego oraz debugowania lub testowania założeń, które robią inne podejścia analityczne