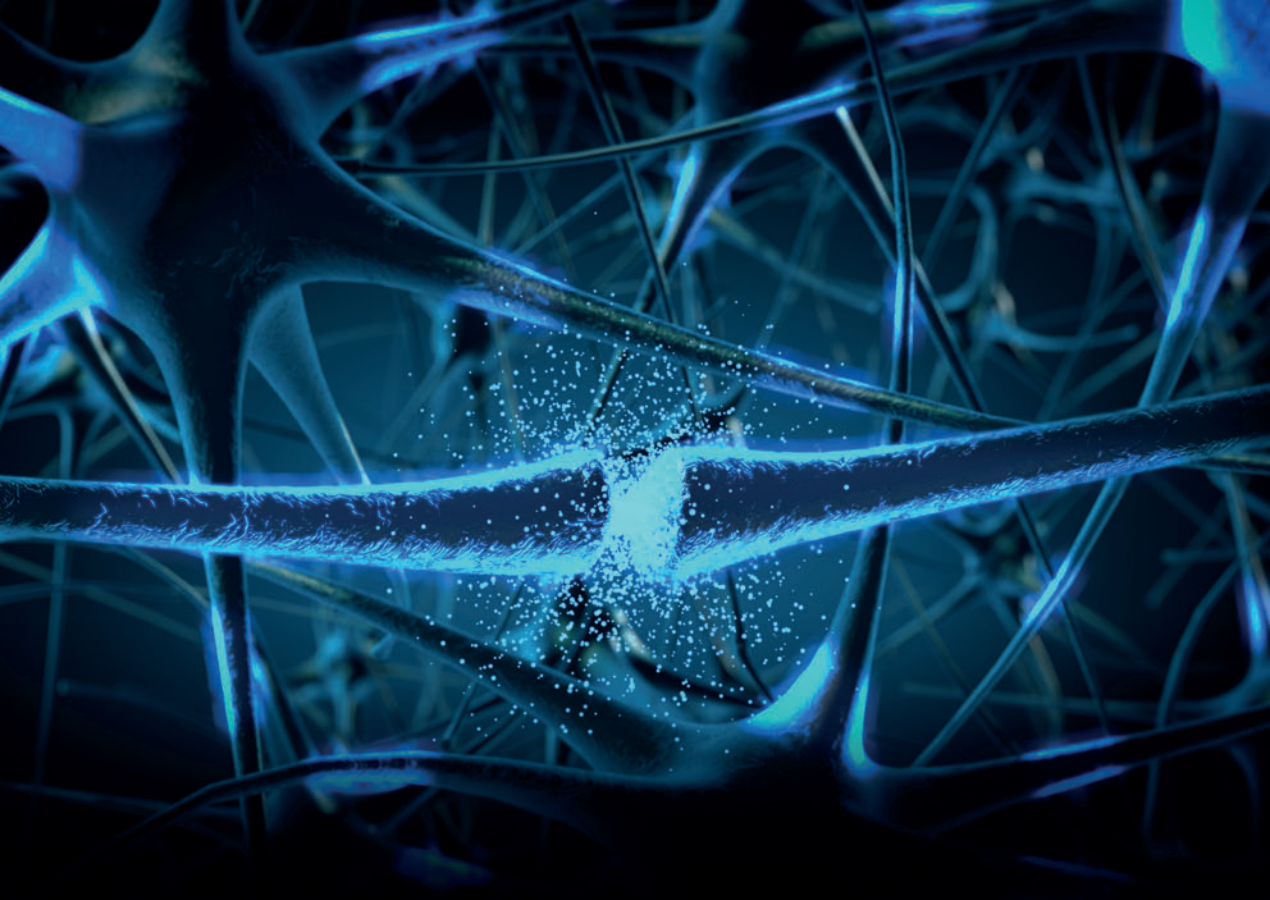


Douglas McIlwraith, Haralambos Marmanis, Dmitry Babenko

INTELIGENTNA SIEĆ

Algorytmy przyszłości

WYDANIE II

Helion

Tytuł oryginału: Algorithms of the Intelligent Web, 2nd Edition

Tłumaczenie: Tomasz Walczak

Projekt okładki: Studio Gravite / Olsztyn; Obarek, Pokoński, Pazdrijowski, Zaprucki
Materiały graficzne na okładce zostały wykorzystane za zgodą Shutterstock Images LLC.

ISBN: 978-83-283-3250-8

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *http://helion.pl* (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
http://helion.pl/user/opinie/intsi2
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:
ftp://ftp.helion.pl/przyklady/intsi2.zip

Printed in Poland.

Spis treści

Przedmowa 9
Wprowadzenie 11
Podziękowania 13
O książce 15

Rozdział 1. Budowanie aplikacji na potrzeby inteligentnej sieci 19

1.1. Inteligentny algorytm w akcji — Google Now 21
1.2. Cykl życia inteligentnych algorytmów 23
1.3. Inne przykłady inteligentnych algorytmów 24
1.4. Czym inteligentne aplikacje nie są 25
1.4.1. Inteligentne algorytmy nie są myślącymi maszynami do uniwersalnych zastosowań 25
1.4.2. Inteligentne algorytmy nie zastąpią ludzi 25
1.4.3. Inteligentne algorytmy nie są odkrywane przez przypadek 26
1.5. Klasy inteligentnych algorytmów 26
1.5.1. Sztuczna inteligencja 27
1.5.2. Uczenie maszynowe 28
1.5.3. Analityka predykcyjna 29
1.6. Ocena działania inteligentnych algorytmów 30
1.6.1. Ocena inteligencji 30
1.6.2. Ocena predykcji 31
1.7. Ważne uwagi na temat inteligentnych algorytmów 33
1.7.1. Dane nie są wiarygodne 34
1.7.2. Wnioskowanie wymaga czasu 34
1.7.3. Wielkość ma znaczenie! 34
1.7.4. Różne algorytmy skalują się w odmienny sposób 35
1.7.5. Nie wszystko jest gwoździem! 35
1.7.6. Dane to nie wszystko 35
1.7.7. Czas treningu może się zmieniać 36
1.7.8. Celem jest generalizacja 36
1.7.9. Ludzka intuicja nie zawsze się sprawdza 36
1.7.10. Pomyśl o zaprojektowaniu nowych cech 36
1.7.11. Poznaj wiele różnych modeli 36
1.7.12. Korelacja nie oznacza związku przyczynowo-skutkowego 37
1.8. Podsumowanie 37

Rozdział 2. Wydobywanie struktury z danych — klastrowanie i transformacja danych 39

2.1. Dane, struktura, błąd systematyczny i szum 41
2.2. „Przekleństwo wymiarów” 44
2.3. Algorytm k-średnich 45
2.3.1. K-średnie w praktyce 49
2.4. Gaussowski model mieszany 52
2.4.1. Czym jest rozkład Gaussa? 52
2.4.2. Maksymalizacja wartości oczekiwanej i rozkład Gaussa 55
2.4.3. Gaussowski model mieszany 55
2.4.4. Przykład uczenia z użyciem gaussowskiego modelu mieszanego 57
2.5. Zależności między k-średnimi i algorytmem GMM 59
2.6. Transformacje osi danych 60
2.6.1. Wektory własne i wartości własne 61
2.6.2. Analiza głównych składowych 61
2.6.3. Przykład zastosowania analizy głównych składowych 63
2.7. Podsumowanie 65

Rozdział 3. Rekomendowanie odpowiednich treści 67

3.1. Wprowadzenie — internetowy sklep z filmami 68
3.2. Odległość i podobieństwo 69
3.2.1. Więcej o odległości i podobieństwie 73
3.2.2. Który wzór na podobieństwo jest najlepszy? 75
3.3. Jak działają systemy rekomendacji? 76
3.4. Filtrowanie kolaboratywne według użytkowników 77
3.5. Rekomendacje według modelu z wykorzystaniem rozkładu SVD 82
3.5.1. Rozkład SVD 83
3.5.2. Rekomendacje z użyciem rozkładu SVD — wybór filmów dla danego użytkownika 84
3.5.3. Rekomendacje z wykorzystaniem rozkładu SVD — określanie użytkowników, których może zainteresować dany film 90
3.6. Konkurs Netflix Prize 93
3.7. Ocenianie systemu rekomendacji 94
3.8. Podsumowanie 96

Rozdział 4. Klasyfikowanie — umieszczanie elementów tam, gdzie ich miejsce 97

4.1. Do czego potrzebna jest klasyfikacja? 98
4.2. Przegląd klasyfikatorów 101
4.2.1. Strukturalne algorytmy klasyfikacji 102
4.2.2. Statystyczne algorytmy klasyfikacji 104
4.2.3. Cykl życia klasyfikatora 105
4.3. Wykrywanie oszustw za pomocą regresji logistycznej 106
4.3.1. Wprowadzenie do regresji liniowej 106
4.3.2. Od regresji liniowej do logistycznej 108
4.3.3. Implementowanie wykrywania oszustw 111

4.4. Czy wyniki są wiarygodne? 119
4.5. Klasyfikowanie w bardzo dużych zbiorach danych 122
4.6. Podsumowanie 124

Rozdział 5. Studium przypadku — prognozowanie kliknięć w reklamie internetowej 127

5.1. Historia i informacje wstępne 128
5.2. Giełda 130
5.2.1. Dopasowywanie plików cookie 130
5.2.2. Oferty 131
5.2.3. Powiadomienie o wygranej (lub przegranej) w licytacji 132
5.2.4. Umieszczanie reklamy 132
5.2.5. Monitorowanie reklam 132
5.3. Czym jest agent? 133
5.3.1. Wymagania stawiane agentowi 133
5.4. Czym jest system podejmowania decyzji? 134
5.4.1. Informacje o użytkowniku 135
5.4.2. Informacje o przestrzeni reklamowej 135
5.4.3. Informacje o kontekście 135
5.4.4. Przygotowywanie danych 135
5.4.5. Model dla systemu podejmowania decyzji 136
5.4.6. Odwzorowywanie prognozowanego współczynnika kliknięć na oferowaną kwotę 136
5.4.7. Inżynieria cech 137
5.4.8. Trening modelu 137
5.5. Predykcja kliknięć za pomocą biblioteki Vowpal Wabbit 138
5.5.1. Format danych używany w VW 138
5.5.2. Przygotowywanie zbioru danych 141
5.5.3. Testowanie modelu 146
5.5.4. Kalibrowanie modelu 148
5.6. Komplikacje związane z budowaniem systemu podejmowania decyzji 150
5.7. Przyszłość prognozowania zdarzeń w czasie rzeczywistym 150
5.8. Podsumowanie 151

Rozdział 6. Uczenie głębokie i sieci neuronowe 153

6.1. Intuicyjne omówienie uczenia głębokiego 154
6.2. Sieci neuronowe 155
6.3. Perceptron 156
6.3.1. Trening 158
6.3.2. Trening perceptronu z użyciem pakietu scikit-learn 160
6.3.3. Geometryczna interpretacja działania perceptronu dla dwóch wejść 162
6.4. Perceptrony wielowarstwowe 164
6.4.1. Trening z wykorzystaniem propagacji wstecznej 167
6.4.2. Funkcje aktywacji 168
6.4.3. Intuicyjne wyjaśnienie propagacji wstecznej 169
6.4.4. Teoria propagacji wstecznej 170
6.4.5. Wielowarstwowe sieci neuronowe w pakiecie scikit-learn 172
6.4.6. Perceptron wielowarstwowy po zakończeniu nauki 174

6.5. Zwiększanie głębokości — od wielowarstwowych sieci neuronowych do uczenia głębokiego 175
6.5.1. Ograniczone maszyny Boltzmanna 176
6.5.2. Maszyny BRBM 177
6.5.3. Maszyny RBM w praktyce 180
6.6. Podsumowanie 183

Rozdział 7. Dokonywanie właściwego wyboru 185

7.1. Testy A/B 187
7.1.1. Teoria 187
7.1.2. Kod 190
7.1.3. Adekwatność testów A/B 191
7.2. Wieloręki bandyta 192
7.2.1. Strategie stosowane w problemie wielorękiego bandyty 192
7.3. Strategia bayesowska w praktyce 197
7.4. Testy A/B a strategia bayesowska 207
7.5. Rozwinięcia eksperymentu z wielorękim bandytą 208
7.5.1. Bandyci kontekstowi 209
7.5.2. Problem bandytów z przeciwnikiem 210
7.6. Podsumowanie 210

Rozdział 8. Przyszłość inteligentnej sieci 213

8.1. Przyszłe zastosowania inteligentnej sieci 214
8.1.1. Internet rzeczy 214
8.1.2. Opieka zdrowotna w domu 215
8.1.3. Autonomiczne samochody 215
8.1.4. Spersonalizowane fizyczne reklamy 216
8.1.5. Sieć semantyczna 216
8.2. Społeczne implikacje rozwoju inteligentnej sieci 217

Dodatek. Pobieranie danych z sieci WWW 219

Przykład — wyświetlanie reklam w internecie 220
Dane dostępne w kontekście reklamy internetowej 220
Rejestrowanie danych — naiwne rozwiązanie 221
Zarządzanie zbieraniem danych w dużej skali 222
Poznaj system Kafka 224
Replikacja w systemie Kafka 226
Grupy konsumentów, równoważenie i kolejność 232
Łączenie wszystkich elementów 233
Ocena systemu Kafka — rejestrowanie danych w dużej skali 236
Wzorce projektowe w systemie Kafka 238
Łączenie systemów Kafka i Storm 238
Łączenie systemów Kafka i Hadoop 240

Skorowidz 243

Przedmowa

Sieć WWW to infrastruktura wykorzystywana przez oparte na internecie społeczeństwo informacyjne. Jest to podstawowe narzędzie, z którego miliardy osób korzystają do interakcji w internecie. Postęp przemysłowy dokonuje się dzięki rozwijaniu usług informacyjnych w internecie. Obecnie dzięki dojrzałym technologiom przetwarzania w chmurze i komunikacji bezprzewodowej sieć WWW staje się nie tylko narzędziem do publikowania i konsumowania informacji, ale też platformą, w której można rozwijać i wdrażać usługi informacyjne, a także udostępniać je miliardom użytkowników w dowolnym miejscu i czasie. Duże zbiory danych (ang. *big data*) zapewniają bogate materiały do budowania wszechstronnych usług, a także umożliwiają wbudowanie inteligencji w usługi i ulepszenie dzięki temu wrażeń z użytkowania usług w sieci WWW. Inteligentne usługi sprawiają, że sieć WWW zmienia nasze życie. Pomaga nam znaleźć odpowiednią restaurację, zaplanować wymarzone wakacje, kupić niemal dowolny produkt i tworzyć społeczności mające najróżniejsze cele. Uzyskanie tej inteligencji jest możliwe dzięki analizie danych wygenerowanych w wyniku interakcji użytkowników z zawartością sieci WWW. Rozwijanie inteligencji w sieci jest więc jednym z podstawowych aspektów nowoczesnej nauki o danych.

Mam wielką przyjemność zaprezentować Ci tę doskonałą książkę, *Inteligentna sieć. Algorytmy przyszłości*, zaktualizowaną przez młodego, ale bardzo doświadczonego badacza danych, dr. Douglasa McIlwraitha. Ta pozycja ma pokazać istotę inteligentnych aplikacji sieciowych — algorytmy zapewniające inteligencję. To ambitny cel. Zaskoczyło mnie to, że Doug zdołał kompleksowo przedstawić ten obszerny temat w przystępnym języku na mniej niż 250 stronach.

W tej książce opisane są najpopularniejsze techniki o szerokim spektrum zastosowań. Znajdziesz tu zwięzły opis algorytmów i ich matematycznych podstaw oraz kod w Pythonie. Lektura tej książki była dla mnie prawdziwą przyjemnością. Mam nadzieję, że Tobie też się ona spodoba. Ważniejsze jest jednak to, że po zakończeniu lektury powinieneś zyskać umiejętności i wiedzę pozwalające zwiększyć inteligencję sieci WWW.

Yike Guo
Profesor i dyrektor
Data Science Institute,
Imperial College, Londyn

1 Budowanie aplikacji na potrzeby inteligentnej sieci

Zawartość rozdziału:

- Dostrzeganie inteligencji w sieci
- Typy inteligentnych algorytmów
- Ocena inteligentnych algorytmów

Określenie „inteligentna sieć" dla różnych osób znaczy coś innego. Dla niektórych związane jest z ewolucją sieci WWW w kierunku reagującego na interakcje i przydatnego bytu, który potrafi uczyć się od użytkowników i reagować na ich zachowania. Dla innych znaczy wkroczenie sieci WWW w wiele aspektów naszego życia. Dla mnie inteligentna sieć jest daleka od pierwszej wersji Skynetu, w którym komputery przejmują władzę w dystopicznej przyszłości. Jest natomiast związana z projektowaniem i implementowaniem w naturalny sposób reagujących aplikacji, dzięki którym wrażenia z użytkowania internetu są w policzalny sposób lepsze. Prawdopodobnie każdy z Czytelników zetknął się w wielu sytuacjach z inteligencją maszynową. W tym rozdziale przedstawiamy przykłady, które ułatwią Ci dostrzeżenie jej w przyszłości. To z kolei pomoże Ci zrozumieć, co tak naprawdę dzieje się na zapleczu, gdy wchodzisz w interakcje z inteligentnymi aplikacjami.

Skoro wiesz już, że ta książka nie dotyczy pisania bytów, które spróbują przejąć kontrolę nad światem, warto wspomnieć też o innych rzeczach pominiętych na jej stronach. Przede wszystkim jest to książka skoncentrowana na technologiach używanych na zapleczu. Nie przeczytasz tu o atrakcyjnych interaktywnych wizualizacjach

lub platformach. Te tematy poznasz dzięki świetnym publikacjom Scotta Murraya[1], Davida McCandlessa[2] i Edwarda Tuftego[3]. Tu nie mamy miejsca na omówienie tego zagadnienia w stopniu, na jaki zasługuje. Z tej książki nie nauczysz się też statystyki. Jednak aby jak najlepiej wykorzystać zawartość tej pozycji, powinieneś znać przynajmniej podstawy tej dziedziny i ukończyć kurs statystyki.

Nie jest to też książka o nauce o danych. Dostępnych jest wiele tytułów pomocnych dla praktyków nauki o danych. Mamy nadzieję, że także ta książka będzie dla nich przydatna, jednak w jej rozdziałach znajdziesz mało szczegółów na temat tego, jak być naukowcem z tej dziedziny. Jej omówienie znajdziesz w tekstach Joela Grusa[4] oraz Fostera Provosta i Toma Fawcetta[5].

Ponadto ta książka nie jest szczegółowym omówieniem projektowania algorytmów. Często pomijamy szczegóły projektowania algorytmów i przedstawiamy intuicyjny opis zamiast wnikania w szczegóły. Pozwala to omówić więcej zagadnień, choć może dziać się to kosztem precyzji. Każdy rozdział możesz traktować jak trop prowadzący przez ważne aspekty danego podejścia i pozwalający dotrzeć do zasobów zawierających więcej szczegółów.

Choć wiele przykładów z tej książki jest napisanych z użyciem pakietu scikit-learn (*http://scikit-learn.org*), nie jest to pozycja poświęcona temu narzędziu. Ten pakiet to tylko narzędzie pozwalające zademonstrować prezentowane w tekście techniki. Każdy przykład opatrzyliśmy przynajmniej ogólnym objaśnieniem tego, dlaczego dany algorytm działa. W niektórych sytuacjach przedstawiamy więcej szczegółów, jednak w wielu miejscach powinieneś kontynuować poszukiwania poza tą książką.

O czym więc jest ta pozycja? Omawiamy tu narzędzia związane z całym procesem funkcjonowania nowoczesnych inteligentnych algorytmów. Opisujemy informacje zbierane na temat przeciętnych użytkowników sieci, które mogą być przetwarzane w przydatne strumienie pozwalające prognozować zachowania tych osób (oraz modyfikować prognozy w reakcji na zmiany zachowań użytkowników). To oznacza, że często odchodzimy od modelu typowego dla książek o podstawach algorytmów i dajemy Ci przedsmak (!) wszystkich ważnych aspektów inteligentnych algorytmów.

Omawiamy nawet (w dodatku) technologię publikuj-subskrybuj, która umożliwia porządkowanie dużych ilości danych w trakcie zbierania. Choć nie jest to temat do książki poświęconej ściśle nauce o danych lub algorytmom, uważamy, że należy opisać go w pozycji dotyczącej inteligentnej sieci. Nie oznacza to, że ignorujemy naukę o danych lub algorytmy — w żadnym razie! Omawiamy tu większość najważniejszych algorytmów używanych przez czołowych graczy w dziedzinie inteligentnych algorytmów.

---

[1] Scott Murray, *Interactive Data Visualization for the Web* (O'Reilly, 2013).

[2] David McCandless, *Information Is Beautiful* (HarperCollins, 2010).

[3] Edward Tufte, *The Visual Display of Quantitative Information* (Graphics Press USA, 2001).

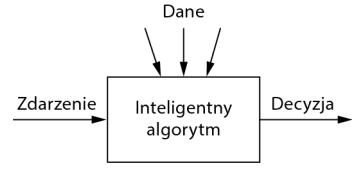
[4] Joel Grus, *Data Science From Scratch: First Principles with Python* (O'Reilly, 2015).

[5] Foster Provost i Tom Fawcett, *Data Science for Business* (O'Reilly Media, 2013).

Tam, gdzie to możliwe, wskazujemy znane przykłady zastosowania technik w praktyce. Dzięki temu będziesz mógł porównać swoją wiedzę z działaniem systemów — i bez wątpienia zrobić tym wrażenie na znajomych!

Wybiegamy jednak za daleko w przyszłość. W tym rozdziale przedstawiamy kilka przykładów zastosowania inteligentnych algorytmów, które powinieneś natychmiast rozpoznać. Opisujemy, czego inteligentne algorytmy nie potrafią, a następnie przedstawiamy taksonomię omawianego obszaru, z którą będziesz mógł wiązać poznawane zagadnienia. W końcowej części prezentujemy szereg metod oceny inteligentnych algorytmów i przedstawiamy kilka przydatnych informacji.

Już słyszymy, jak pytasz: „Czym jest inteligentny algorytm?". Na potrzeby tej książki za *inteligentny* uznajemy każdy algorytm, który wykorzystuje dane do modyfikacji swojego działania. Pamiętaj, że gdy wchodzisz w interakcje z algorytmem, komunikujesz się wyłącznie z zestawem określonych reguł. Inteligentne algorytmy różnią się od innych tym, że mogą zmieniać swoje działanie w trakcie pracy. Często użytkownik ma wrażenie, że taki algorytm jest inteligentny. Rysunek 1.1 przedstawia takie algorytmy. Widać tu, że inteligentny algorytm reaguje na zachodzące w środowisku zdarzenia i podejmuje decyzje. Zbierając dane (mogą nimi być też same zdarzenia) z kontekstu, w którym działa, algorytm ewoluuje — w tym sensie, że decyzje nie zależą deterministycznie od samego zdarzenia. Inteligentny algorytm w różnych momentach może podejmować odmienne decyzje w zależności od zebranych danych.



Rysunek 1.1. Ogólny obraz pracy inteligentnego algorytmu. Taki algorytm przejawia inteligencję, ponieważ podejmuje decyzje na podstawie zebranych danych

1.1. Inteligentny algorytm w akcji — Google Now

Aby zilustrować proces pokazany na rysunku, postaramy się przeprowadzić analizę aplikacji Google Now. Warto wspomnieć, że jej szczegóły są chronione przez firmę Google, dlatego wykorzystujemy nasze doświadczenie do pokazania, jak algorytm z tej aplikacji może działać na zapleczu.

Użytkownicy urządzeń z systemem Android zapewne natychmiast rozpoznają ten produkt, a dla osób korzystających z systemu iOS mamy informację, że Google Now to odpowiedź Google'a na program Siri. Google reklamuje go za pomocą hasła: „Odpowiednie informacje we właściwym czasie". Google Now to aplikacja potrafiąca wykorzystać różne źródła informacji i powiadamiać użytkowników o pobliskich restauracjach, wydarzeniach, korkach i podobnych rzeczach, które uzna za interesujące dla danej osoby. Aby pokazać, czym jest inteligentny algorytm, posłużymy się konkretnym przykładem z aplikacji Google Now. Gdy wykryje ona korek na standardowej drodze użytkownika do pracy, wyświetla określone informacje przed wyjściem danej osoby z domu. Świetnie! Jak jednak jest to możliwe?

Zacznijmy od zrozumienia, co się tu dzieje. Aplikacja zna lokalizację użytkownika dzięki modułowi GPS i zarejestrowanym stacjom łączności bezprzewodowej. Dlatego aplikacja zawsze wie, gdzie użytkownik się znajduje (z dość dużą dokładnością).

W kontekście rysunku 1.1 jest to jeden z aspektów danych używanych do zmiany działania algorytmu. Teraz potrzebny jest tylko niewielki krok, by ustalić lokalizację domu i pracy. Odbywa się to dzięki wykorzystaniu *uprzedniej wiedzy*, która została wbudowana w algorytm, zanim zaczął on uczyć się na podstawie danych. Tu uprzednia wiedza może mieć postać następujących reguł:

- Lokalizacja najczęściej wykrywana w nocy to dom.
- Lokalizacja najczęściej wykrywana w ciągu dnia to praca.
- Ludzie (w większości) prawie każdego dnia jadą do pracy, a następnie z powrotem do domu.

Choć ten przykład nie jest idealny, dobrze ilustruje pewną kwestię: w społeczeństwie używane są pojęcia „pracy", „domu" i „dojazdów", a na podstawie danych i *modelu* można wyciągać wnioski. Tu można ustalić prawdopodobną lokalizację domu i pracy wraz z prawdopodobnymi trasami dojazdu. Używamy tu określenia *prawdopodobne*, ponieważ w wielu modelach uwzględniane jest prawdopodobieństwo określonych decyzji.

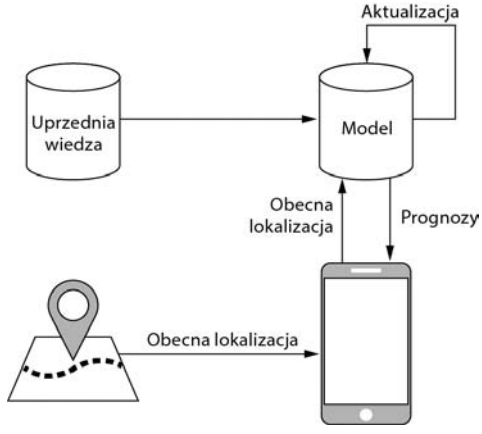
Gdy użytkownik kupuje nowy telefon lub rejestruje nowe konto w usługach firmy Google, Google Now potrzebuje czasu na wyciągnięcie wniosków. Podobnie dzieje się, gdy użytkownik zmieni mieszkanie lub pracę. Aplikacja musi wtedy nauczyć się nowych lokalizacji. Szybkość reagowania modelu na zmiany to *szybkość uczenia się*.

Nadal jednak brakuje informacji, aby móc wyświetlać adekwatne informacje dotyczące trasy dojazdu (aby podejmować *decyzje* na podstawie *zdarzeń*). Ostatni fragment układanki wymaga prognozowania, kiedy użytkownik zamierza opuścić jedną lokalizację i wyruszyć w drogę do drugiej. Podobnie jak wcześniej można utworzyć model i określić godzinę dojazdów, a następnie aktualizować ją, aby odzwierciedlić zmiany wzorców zachowania. W przyszłości można określić prawdopodobieństwo, z jakim użytkownik znajduje się w danej lokalizacji i szykuje się do drogi. Jeśli to prawdopodobieństwo przekroczy poziom progowy, Google Now może sprawdzić informacje o korkach i przekazać je w powiadomieniu użytkownikowi.

Ten konkretny aspekt aplikacji Google Now jest dość skomplikowany i prawdopodobnie pracuje nad nim specjalny zespół. Jednak łatwo jest zauważyć, że schemat działania tego narzędzia oparty jest na inteligentnym algorytmie. Aplikacja wykorzystuje *dane* na temat dojazdów, aby zrozumieć zwyczaje użytkownika i przygotować dla niego spersonalizowane podpowiedzi (*decyzje*) na podstawie obecnej lokalizacji (*zdarzenie*). Rysunek 1.2 przedstawia ten proces w formie graficznej.

Warto zauważyć, że produkt Google Now prawdopodobnie wykorzystuje na zapleczu cały pakiet inteligentnych algorytmów. Przeprowadzają one przeszukiwanie tekstowe kalendarza Google, starając się zrozumieć plan dnia użytkownika, a modele badania zainteresowań próbują ustalić, które wyniki wyszukiwania są przydatne i czy należy oznaczyć nowe treści jako interesujące.

Programista inteligentnych algorytmów musi korzystać ze swoich umiejętności do budowania nowych rozwiązań na podstawie złożonych wymagań i starannie identyfikować wszystkie podzadania, które można wykonać za pomocą istniejącej klasy takich

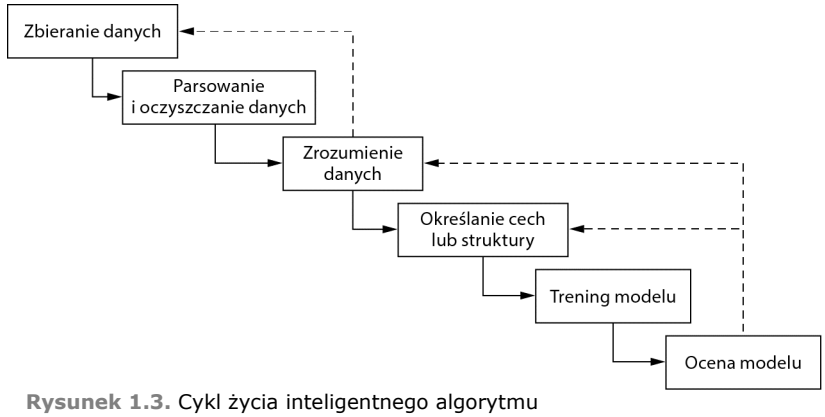


Rysunek 1.2. Graficzne ujęcie jednego z aspektów projektu Google Now. Aby aplikacja Google Now mogła prognozować przyszłe lokalizacje, używa modelu uwzględniającego przeszłe lokalizacje i obecną pozycję. Uprzednia wiedza pozwala wbudować wcześniej znane informacje w system

algorytmów. Każde tworzone rozwiązanie powinno być oparte na dokonaniach z omawianej dziedziny i zbudowane na ich podstawie. Wiele takich dokonań omawiamy w tej książce. Wprowadziliśmy tu kilka ważnych pojęć wyróżnionych kursywą. Używamy ich w dalszych rozdziałach przy szczegółowym omawianiu poszczególnych algorytmów.

1.2. Cykl życia inteligentnych algorytmów

W poprzednim podrozdziale opisaliśmy, że inteligentny algorytm składa się z czarnej skrzynki, przyjmuje dane i generuje prognozy na podstawie zdarzeń. Przedstawiliśmy konkretny przykład z firmy Google w postaci projektu Google Now. Może się zastanawiasz, jak projektanci inteligentnych algorytmów dochodzą do rozwiązań. Istnieje ogólny cykl życia zaadaptowany z książki *Computational Information Design* Bena Fry'a[6]. Możesz posługiwać się tym cyklem w trakcie projektowania własnych rozwiązań. Cykl ten pokazany jest na rysunku 1.3.



Rysunek 1.3. Cykl życia inteligentnego algorytmu

[6] Ben Fry, praca doktorska, *Computational Information Design* (MIT, 2004).

Gdy projektujesz inteligentne algorytmy, najpierw musisz pomyśleć o zbieraniu danych (na tym koncentrujemy się w dodatku), a następnie zająć się ich parsowaniem i oczyszczaniem, ponieważ ich format jest niewłaściwy. Następnie trzeba zrozumieć dane, co można uzyskać dzięki ich eksploracji i wizualizacji. Następnie możesz przedstawić dane w odpowiednich formatach (omawiamy to w rozdziale 2.). Na tym etapie jesteś gotowy do rozpoczęcia treningu modelu i oceny zdolności predykcyjnej utworzonego rozwiązania. W rozdziałach od 3. do 7. omawiamy różne modele, którymi możesz się posługiwać. Po zakończeniu każdego etapu możesz cofnąć się do wcześniejszych kroków. Najczęściej używane ścieżki powrotne są pokazane za pomocą przerywanych linii na rysunku 1.3.

1.3. Inne przykłady inteligentnych algorytmów

Przyjrzyjmy się innym aplikacjom z ostatniej dekady, w których też używana jest inteligencja oparta na algorytmach. Punktem zwrotnym w historii sieci WWW było pojawienie się wyszukiwarek. Jednak duża część możliwości sieci WWW pozostawała niewykorzystana aż do 1998 roku, kiedy to zaczęto analizować odsyłacze w kontekście wyszukiwania. Od tego czasu w niecałe 20 lat firma Google rozwinęła się od startupu do lidera w branży technologicznej. Początkowo rozwój tej firmy wynikał z sukcesu wyszukiwania opartego na odsyłaczach, a później — z licznych nowych i innowacyjnych aplikacji z obszaru usług mobilnych i działających w chmurze.

Jednak świat inteligentnych aplikacji sieciowych nie ogranicza się do wyszukiwarek. Amazon był jednym z pierwszych sklepów internetowych, w których użytkownikom prezentowano rekomendacje oparte na wzorcach zakupowych. Możliwe, że znasz tę funkcję. Załóżmy, że kupujesz książkę na temat platformy JavaServer Faces i inną o Pythonie. Gdy tylko dodasz je do koszyka zakupów, Amazon zaproponuje dodatkowe pozycje powiązane z tymi, które już wybrałeś. Możliwe, że będą to książki dotyczące AJAX-a lub Ruby on Rails. Ponadto gdy ponownie otworzysz witrynę Amazonu, może ona zarekomendować te same lub inne powiązane produkty. Inną inteligentną aplikacją sieciową jest Netflix — największy na świecie internetowy serwis strumieniowania filmów. Oferuje on ponad 53 milionom subskrybentów dostęp do stale zmieniającej się biblioteki filmów i seriali, które można natychmiast obejrzeć za pomocą technologii strumieniowania.

Sukces Netfliksa wynika po części z zapewnienia użytkownikom łatwego sposobu wyboru filmów z bogatej kolekcji. Odpowiada za to system rekomendacji Cinematch. Jego zadanie polega na prognozowaniu, czy użytkownikowi spodoba się dany film. Uwzględniane jest przy tym to, czy danej osobie spodobały się inne filmy. Jest to następny świetny przykład inteligentnej aplikacji sieciowej. Zdolność predykcyjna algorytmu Cinematch jest dla Netfliksa tak ważna, że w październiku 2006 roku firma ogłosiła konkurs na jego usprawnienie z główną nagrodą miliona dolarów. We wrześniu 2009 roku nagroda została przyznana zespołowi BellKor's Pragmatic Chaos. W rozdziale 3. omawiamy algorytmy potrzebne do budowania systemów rekomendacji takich jak Cinematch oraz opisujemy zwycięski projekt.

Wykorzystywanie opinii społeczności do generowania inteligentnych prognoz nie ogranicza się do rekomendacji książek lub filmów. Firma PredictWallStreet rejestruje prognozy użytkowników dotyczące określonych akcji lub indeksów, aby wykryć trendy w opiniach graczy giełdowych i przewidzieć wartość danych papierów wartościowych. Nie zalecamy podjęcia wszystkich oszczędności i rozpoczęcia inwestowania zgodnie z prognozami tej firmy, jest to jednak następny przykład kreatywnego zastosowania w praktyce technik omawianych w tej książce.

1.4. Czym inteligentne aplikacje nie są

Ponieważ w sieci WWW działa tak wiele inteligentnych algorytmów, łatwo jest dojść do wniosku, że odpowiednia liczba inżynierów zdoła opracować lub zautomatyzować dowolny proces. Niech jednak powszechność takich rozwiązań Cię nie zwiedzie.

Każda odpowiednio zaawansowana technologia jest nieodróżnialna od magii.

— Arthur C. Clarke

Po zapoznaniu się z aplikacją Google Now możesz mieć skłonność do podejrzewania bardziej rozbudowanych aplikacji o większą inteligencję. Jednak w rzeczywistości takie aplikacje łączą zestaw uczących się algorytmów w celu zapewnienia przydatnego rozwiązania, ograniczonego jednak do konkretnych problemów. Dlatego nie czuj się przytłoczony złożonością zadania, ale zadaj sobie pytanie: „Które aspekty problemu są możliwe do wyuczenia i zamodelowania?". Dopiero potem będziesz mógł opracować rozwiązania przejawiające inteligencję. W dalszych podrozdziałach omawiamy najczęściej przyjmowane błędne założenia dotyczące inteligentnych algorytmów.

1.4.1. Inteligentne algorytmy nie są myślącymi maszynami do uniwersalnych zastosowań

Na początku tego rozdziału wspomnieliśmy, że nie jest to książka poświęcona budowaniu czujących istot. Omawiamy tu budowanie algorytmów, które potrafią dostosować swoje działanie na podstawie otrzymanych danych. Zgodnie z naszym doświadczeniem projekty biznesowe, które najczęściej kończą się niepowodzeniem, to te z tak rozbudowanymi celami jak rozwiązanie całego problemu sztucznej inteligencji! Zacznij od czegoś prostego i rozwijaj to, stale oceniając aplikację do momentu, w którym uznasz, że pierwotny problem został rozwiązany.

1.4.2. Inteligentne algorytmy nie zastąpią ludzi

Inteligentne algorytmy świetnie radzą sobie z uczeniem się konkretnego zagadnienia na podstawie odpowiednich danych. Nie są jednak dobre w uczeniu się nowych zagadnień wykraczających poza to, jak zostały zaprogramowane. Dlatego inteligentne algorytmy i rozwiązania trzeba opracowywać i łączyć w staranny sposób, aby zapewnić zadowalające efekty.

Ludzie natomiast są doskonałymi uniwersalnymi maszynami obliczeniowymi. Potrafią łatwo zrozumieć nowe koncepcje i wykorzystać wiedzę z jednej dziedziny w innej.

Posiadają różne serwomechanizmy (!) i można ich programować w wielu różnych językach (!!). Błędem jest myśleć, że można łatwo napisać oparte na kodzie rozwiązania wykonujące pozornie proste ludzkie czynności.

Wiele wymagających udziału człowieka procesów w firmach i organizacjach na pozór wydaje się prostych, jednak zwykle wynika to z tego, że kompletna specyfikacja danego procesu jest nieznana. Dalsze analizy zwykle prowadzą do wykrycia rozbudowanej komunikacji z użyciem różnych kanałów i często także koniecznością uwzględnienia sprzecznych celów. Inteligentne algorytmy słabo sobie radzą w takich scenariuszach i wymagają uproszczenia oraz sformalizowania procesu.

Warto przedstawić prostą, ale trafną analogię do automatyzacji linii montażowych pojazdów silnikowych. W porównaniu z początkami automatyzacji w tej dziedzinie (gdzie pionierem był Henry Ford) obecnie można całkowicie zautomatyzować kroki procesu produkcji z użyciem robotów. Nie zostało to uzyskane, jak mógłby to sobie wyobrażać Henry Ford, dzięki zbudowaniu uniwersalnych humanoidalnych robotów, które zastąpiły pracowników. Automatyzacja była możliwa dzięki abstrakcyjnemu przedstawieniu linii montażowej i rygorystycznemu sformalizowaniu procesu. To z kolei doprowadziło do ścisłego zdefiniowania podzadań, które *można* było rozwiązać dzięki robotyzacji. Choć teoretycznie możliwe jest opracowanie zautomatyzowanych procesów uczenia na potrzeby ręcznie wykonywanych optymalizacji, wymagałoby to podobnego przeprojektowania i formalizacji zadań.

1.4.3. Inteligentne algorytmy nie są odkrywane przez przypadek

Najlepsze inteligentne algorytmy są często wynikiem wykorzystania prostych abstrakcji i mechanizmów. Takie algorytmy wyglądają na skomplikowane, ponieważ uczą się i zmieniają, ale mechanizmy, na których są oparte, są proste. Natomiast inteligentne algorytmy niskiej jakości często są oparte na wielu warstwach skomplikowanych reguł dodawanych niezależnie od siebie w celu rozwiązania konkretnych sytuacji. Ujmijmy to tak: zawsze zaczynaj od możliwie najprostszego modelu. Następnie staraj się stopniowo uzyskiwać lepsze wyniki, wbudowując w rozwiązanie dodatkowe inteligentne aspekty. Reguła **KISS** (ang. *keep it simple, stupid*, czyli nie komplikuj, głupku) jest Twoim przyjacielem i niezmienną zasadą inżynierii oprogramowania.

1.5. Klasy inteligentnych algorytmów

Może pamiętasz, że posłużyliśmy się nazwą *inteligentny algorytm* do opisu dowolnego algorytmu, który może modyfikować swoje działanie na podstawie danych. W tej książce to bardzo ogólne określenie ma obejmować wszystkie aspekty inteligencji i uczenia się. Gdy zajrzysz do innych pozycji, prawdopodobnie natrafisz na odmienne określenia, które po części się pokrywają. Oto one: *uczenie maszynowe* (ang. *machine learning* — **ML**), analityka predykcyjna (ang. *predictive analytics* — **PA**) i *sztuczna inteligencja* (ang. *artificial intelligence* — **AI**). Rysunek 1.4 przedstawia zależności między tymi dziedzinami.

Choć we wszystkich trzech dziedzinach występują algorytmy wykorzystujące dane do modyfikowania działania, w każdym z tych obszarów kluczowe są inne aspekty. W następnych podpunktach omawiamy po kolei każdy z tych obszarów, aby zapewnić Ci wiedzę niezbędną do powiązania tych dziedzin.

Rysunek 1.4. Taksonomia inteligentnych algorytmów

1.5.1. Sztuczna inteligencja

Sztuczna inteligencja, powszechnie znana pod akronimem AI, powstała jako dziedzina informatyki około 1950 roku. Początkowo badacze sztucznej inteligencji mieli ambitne plany i chcieli opracować maszyny myślące podobnie jak ludzie[7]. Z czasem, gdy ustalono pełen zakres prac potrzebnych do zasymulowania inteligencji, cele badaczy stały się bardziej praktyczne i konkretne. Obecnie stosowanych jest wiele definicji sztucznej inteligencji. Na przykład Stuart Russell i Peter Norvig opisują ją tak: „Dziedzina badań nad agentami, które przyjmują bodźce ze środowiska i wykonują działania”[8], natomiast John McCarthy stosuje następującą definicję: „Dziedzina nauki i inżynierii związana z budowaniem inteligentnych maszyn, a zwłaszcza inteligentnych programów komputerowych”[9]. McCarthy dodaj też, że: „Inteligencja jest obliczeniowym aspektem możliwości realizowania celów w świecie”.

W większości omówień sztuczna inteligencja wiązana jest z badaniami nad agentami (oprogramowaniem i maszynami), które mają zestaw opcji do wyboru i muszą zrealizować konkretne cele. Badania dotyczą określonych dziedzin problemowych (na przykład gier w go[10], szachy[11] i Jeopardy![12]) i w takich ograniczonych środowiskach efekty są często znakomite. Na przykład komputer Deep Blue firmy IBM w 1997 roku pokonał Garriego Kasparowa w szachy, a w 2011 roku komputer Watson tej samej firmy wygrał pierwszą nagrodę miliona dolarów w amerykańskim teleturnieju Jeopardy! Niestety, nieliczne algorytmy dobrze radzą sobie w wymyślonej przez Alana Turinga *grze w naśladowanie*[13] (nazywanej też testem Turinga), uznawanej przez większość autorów za standardowy test na inteligencję. W tej grze chodzi o to, aby sędzia nie potrafił wykryć, że jego rozmówca jest maszyną (eliminowane są przy tym wskazówki wizualne

[7] Herbert Simon, *The Shape of Automation for Men and Management* (Harper & Row, 1965).

[8] Stuart Russell i Peter Norvig, *Artificial Intelligence: A Modern Approach* (Prentice Hall, 1994).

[9] John McCarthy, *What Is Artificial Intelligence?* (Stanford University, 2007), *http://www-formal.stanford.edu/jmc/whatisai*.

[10] Bruno Bouzy i Tristan Cazenave, Computer Go: An AI Oriented Survey, „Artificial Intelligence” (Elsevier) 132, nr 1 (2001): 39 – 103.

[11] Murray Campbell, A.J. Hoane i Feng-hsiung Hsu, *Deep Blue*, „Artificial Intelligence” (Elsevier) 134, nr 1 (2002): 57 – 83.

[12] D. Ferrucci i współpracownicy, *Building Watson: An Overview of the DeepQA Project*, „AI Magazine” 31, nr 3 (2010).

[13] Alan Turing, *Computing Machinery and Intelligence*, „Mind” 59, nr 236 (1950): 433 – 60.

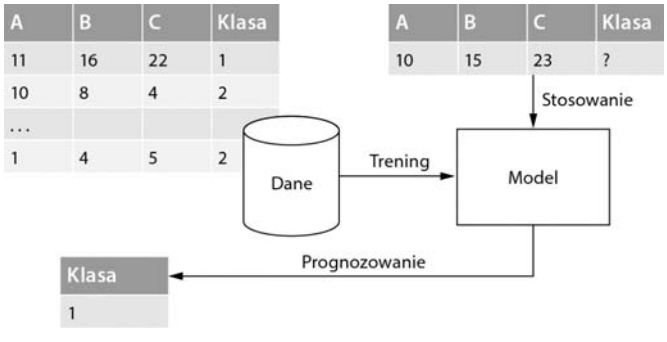
i dźwiękowe — komunikacja odbywa się wyłącznie za pomocą wpisywanego tekstu). To zadanie jest znacznie trudniejsze, ponieważ maszyna musi mieć obszerną wiedzę z wielu obszarów. Sędzia nie jest tu ograniczony, jeśli chodzi o pytania, jakie może zadawać.

1.5.2. Uczenie maszynowe

Uczenie maszynowe związane jest ze zdolnością oprogramowania do generalizowania na podstawie wcześniejszych doświadczeń. Ważne jest to, że te generalizacje mają pozwalać na udzielanie odpowiedzi na pytania dotyczące zarówno wcześniej zebranych danych, jak i nowych informacji. Niektóre techniki uczenia polegają na tworzeniu możliwych do wyjaśnienia modeli — nawet laik może prześledzić proces generalizowania. Przykładami są tu drzewa decyzyjne i, w bardziej ogólnym ujęciu, dowolne metody uczenia oparte na regułach. Jednak inne algorytmy nie są równie transparentne dla ludzi. Do tej kategorii należą sieci neuronowe i maszyny **SVM** (ang. *support vector machines*).

Możesz uznać, że zakres uczenia maszynowego znacznie różni się od tematyki sztucznej inteligencji. W sztucznej inteligencji istotne są *agenty*, które mają realizować *cele* (podobnie jak agent będący człowiekiem działa na ogólnym poziomie w swoim środowisku), natomiast w uczeniu maszynowym ważne są uczenie się i generalizacja (bardziej przypomina to wewnętrzne funkcjonowanie ludzi). W uczeniu maszynowym rozwiązywane są takie problemy jak klasyfikowanie (rozpoznawanie klas na podstawie danych) i regresja (prognozowanie jednego wyniku na podstawie innego).

Na ogólnym poziomie praktycy z dziedziny uczenia maszynowego używają *danych treningowych* do opracowania *modelu*. Ten model generalizuje w pewien sposób (zależny od używanego modelu) relacje między danymi, aby możliwe było generowanie prognoz na temat nienapotkanych wcześniej danych. Ilustruje to rysunek 1.5. W tym przykładzie dane obejmują trzy *cechy*: A, B i C. Cechy to aspekty danych. Jeśli klasy to „kobiety" i „mężczyźni", a zadanie polega na podziale grupy na te klasy, można wykorzystać cechy takie jak wzrost, waga i numer buta.



Rysunek 1.5. Przepływ danych w uczeniu maszynowym. Dane służą do treningu modelu, który można następnie zastosować do nowych danych. Tu schemat ilustruje klasyfikowanie. Schematy obrazujące klastrowanie i regresję wyglądają podobnie

W danych treningowych relacje między *cechami* i *klasami* są znane. W modelu należy ująć te relacje. Po zakończeniu treningu model można zastosować do nowych danych, których klasa jest nieznana.

1.5.3. Analityka predykcyjna

Analityka predykcyjna nie jest tak szeroko opisywana w literaturze akademickiej jak sztuczna inteligencja i uczenie maszynowe. Jednak wraz z dojrzewaniem architektur przetwarzania dużych zbiorów danych i rosnącym apetytem na operacyjne wykorzystanie danych i uzyskanie dzięki nim dodatkowej wartości dziedzina ta zyskuje na popularności. Na potrzeby tej książki używamy przedstawionej poniżej definicji, rozbudowującej definicję *analityki* ze słownika oksfordzkiego. Dodany został fragment wyróżniony kursywą:

> Analityka predykcyjna: systematyczna obliczeniowa analiza danych lub statystyk *w celu tworzenia modeli predykcyjnych*.

Możesz zadać pytanie: „Jak różni się to od uczenia maszynowego, które też dotyczy predykcji?". To dobre pytanie. Ogólnie techniki uczenia maszynowego mają pomóc zrozumieć i zgeneralizować strukturę oraz relacje w zbiorze danych. W analityce predykcyjnej ważne jest generowanie ocen, rankingów i predykcji dotyczących przyszłych danych i trendów, często w środowisku biznesowym lub operacyjnym. Choć to porównanie może wydawać się niejasne, warto zauważyć, że omawiane tu klasy inteligentnych algorytmów w dużym stopniu się pokrywają oraz nie są konkretne i ścisłe.

Co ciekawe, analitycy zwykle nie tworzą rozwiązań z obszaru analityki predykcyjnej. W analityce predykcyjnej ważne jest tworzenie modeli, które na podstawie informacji potrafią reagować szybko i wydajnie, generując przydatne dane wyjściowe prognozujące przyszłe zjawiska. Takie systemy często są tworzone przez inżynierów oprogramowania i naukowców zajmujących się danymi. Sytuację dodatkowo komplikuje to, że w modelach analityki predykcyjnej używane są czasem techniki uczenia maszynowego i sztucznej inteligencji!

PRZYKŁADY Z OBSZARU ANALITYKI PREDYKCYJNEJ

Aby pomóc Ci intuicyjnie zrozumieć ten obszar, przedstawiamy kilka przykładowych rozwiązań z dziedziny analityki predykcyjnej. Pierwszy pochodzi ze świata reklamy internetowej. Uważni użytkownicy internetu zapewne zauważyli, że reklamy często „podążają za nimi", gdy przeglądają różne witryny. Jeśli wcześniej oglądałeś buty na stronie sklepu Nike, na innych stronach często zobaczysz reklamy tego właśnie obuwia! Jest to tak zwany *retargeting*. Za każdym razem, gdy wczytywana jest strona z reklamami, wiele różnych jednostek podejmuje setki decyzji, chcąc wyświetlić Ci określone reklamy. System wymiany reklamy działa w ten sposób, że każda jednostka podaje cenę, jaką jest gotowa zapłacić za wyświetlenie Ci reklamy. Oferent najwyższej kwoty wygrywa prawo do pokazania reklamy. Ponieważ cały proces musi zachodzić w ciągu milisekund, cenę ustala inteligentny algorytm starający się przewidzieć lub ocenić wartość danego użytkownika. To rozwiązanie z obszaru analityki predykcyjnej podejmuje decyzje na podstawie wcześniejszych zachowań użytkownika i daje korzyści w porównaniu z losowym doborem odbiorców reklam. Do tego przykładu wrócimy w rozdziale 5., gdzie zobaczysz, jak problem ten rozwiązało wiele firm reklamujących się w internecie.

Drugi przykład pochodzi z dziedziny kredytów konsumpcyjnych. Za każdym razem, gdy starasz się o kredyt na karcie stałego klienta, karcie kredytowej, u dostawcy telefonii komórkowej lub o kredyt hipoteczny, sprzedawca naraża się na pewne ryzyko, a także może odnieść określone korzyści. Aby zrównoważyć te aspekty, sprzedawcy chcą wiedzieć, że udzielają kredytów zwłaszcza godnym zaufania osobom, a odmawiają klientom, którzy z większym prawdopodobieństwem nie będą spłacać długu. W praktyce podejmowanie takich decyzji zlecane jest wyspecjalizowanym agencjom ratingowym, które za opłatą przekazują sprzedawcy informacje o zdolności kredytowej danej osoby. Ocena jest generowana przez rozwiązanie z obszaru analityki predykcyjnej na podstawie danych historycznych dla danej populacji. Ta ocena to liczba wysoce skorelowana z ryzykiem dotyczącym danej osoby. Im wyższy wynik, tym bardziej godny zaufania jest klient i tym mniejsze ryzyko niespłacenia kredytu. Zauważ, że to podejście daje tylko przewagę statystyczną, ponieważ osoby o wysokiej ocenie kredytowej też mogą zaprzestać spłat (choć — jeśli model działa — zdarza się to rzadziej niż w przypadku klientów o niższej ocenie).

1.6. Ocena działania inteligentnych algorytmów

Do tej pory pisaliśmy o ogólnych klasach inteligentnych algorytmów i przedstawiliśmy kilka przykładów. Jak jednak praktyk z tej dziedziny może ocenić swój algorytm? Ma to duże znaczenie i to z kilku przyczyn. Po pierwsze, bez obiektywnej oceny nie da się śledzić wyników i ustalić, czy modyfikacje ulepszyły rozwiązanie. Po drugie, jeśli nie da się mierzyć wyników, trudno jest uzasadnić sens stosowania rozwiązania. W kontekście biznesowym menedżerowie i technologowie zawsze będą starali się uwzględniać zyski i koszty, a możliwość solidnej oceny rozwiązania pomaga zachować je w środowisku produkcyjnym.

Dalej wracamy do poszczególnych klas inteligentnych algorytmów i omawiamy strategie ich oceny. Choć poruszamy tu kwestię oceny inteligencji, ten podrozdział dotyczy głównie oceny predykcji (związanych z uczeniem maszynowym i analityką predykcyjną). Ocena i definiowanie inteligencji to obszerne tematy, które zasługują na odrębną książkę. Dlatego zamiast omawiać je w tym miejscu, odsyłamy Czytelników do książek Lindy Gottfredson[14], Jamesa Flynna[15] i Alana Turinga[16].

1.6.1. Ocena inteligencji

Wcześniej wspomnieliśmy o teście Turinga. Czasem jednak trzeba ocenić systemy, którym stawiane są mniej ambitne cele — na przykład inteligentne systemy grające w szachy lub biorące udział w teleturnieju Jeopardy! Takie systemy nie potrafią naśla-

[14] Linda S. Gottfredson, *Mainstream Science on Intelligence:An Editorial with 52 Signatories, History, and Bibliography*, „Wall Street Journal", 13 grudnia 1994.

[15] James R. Flynn, *What Is Intelligence? Beyond the Flynn Effect* (Cambridge University Press, 2009).

[16] Alan Turing, *Computing Machinery and Intelligence*.

dować człowieka, ale świetnie radzą sobie w pojedynczych zadaniach. Zaproponowany przez Sandeepa Rajaniego sposób oceny sztucznej inteligencji[17] obejmuje cztery poziomy:

- *Optymalny* — niemożliwe jest uzyskanie lepszych wyników.
- *Zdecydowanie lepszy od człowieka* — działanie lepsze niż jakiegokolwiek człowieka.
- *Lepszy od człowieka* — działanie lepsze niż większości ludzi.
- *Gorszy od człowieka* — działanie gorsze niż większości ludzi.

Na przykład obecny poziom sztucznej inteligencji pozwala tworzyć systemy optymalne w grze w kółko i krzyżyk, lepsze od człowieka (lub nawet zdecydowanie lepsze od człowieka) w szachach i gorsze od człowieka w tłumaczeniu tekstów w językach naturalnych.

1.6.2. Ocena predykcji

Choć sztuczna inteligencja jest interesująca, większość rozwiązań z tej książki nie dotyczy tej dziedziny. Dlatego należy poszukać bardziej adekwatnych sposobów oceny. Pamiętaj, że w uczeniu maszynowym i analityce predykcyjnej celem jest generowanie prognoz na podstawie danych oraz relacji między cechami i docelowymi wartościami (klasami). Istnieje więc konkretny schemat oceny i można zastosować statystykę do formalnego pomiaru wyników.

W tabeli 1.1 przedstawiony jest (skrajnie uproszczony) zbiór danych, który posłuży do zilustrowania pomiaru skuteczności predyktora. Cechy danych są opisane za pomocą liter alfabetu, a obok przedstawione są wartości logiczne oznaczające rzeczywisty i prognozowany wynik. Przyjmij, że predykcje zostały wygenerowane na podstawie zbioru danych testowych, których pierwotny model nie znał. W zbiorze danych testowych rzeczywiste wyniki były ukryte, dlatego model musiał ustalić dane wyjściowe wyłącznie na podstawie cech.

Tabela 1.1. Przykładowy zbiór danych używany do przedstawienia sposobu oceny inteligentnego algorytmu

A	B	...	Rzeczywisty wynik	Predykcja
10	4	...	Prawda	Fałsz
20	7	...	Prawda	Prawda
5	6	...	Fałsz	Fałsz
1	2	...	Fałsz	Prawda

Od razu widać, że do oceny działania tego klasyfikatora można zastosować kilka prostych miar. Przede wszystkim można badać współczynnik predykcji prawdziwie pozytywnych (ang. *true positive rate* — **TPR**), czyli łączną liczbę predykcji prawdziwie pozytywnych podzieloną przez łączną liczbę wartości pozytywnych w całym zbiorze danych. Miarę tę nazywa się czasem *czułością* (ang. *sensivity* lub *recall*). Zauważ jednak, że jest to tylko połowa obrazu! Jeśli zbudujesz klasyfikator, który zawsze generuje

[17] Sandeep Rajani, *Artificial Intelligence — Man or Machine*, „International Journal of Information Technology and Knowledge Management" 4, nr 1 (2011): 173 – 76.

pozytywną prognozę (niezależnie od cech w danych), uzyskasz bardzo wysoką czułość. Dlatego tę miarę trzeba analizować razem z innym wskaźnikiem — *swoistością* (ang. *specificity*; inaczej współczynnik predykcji prawdziwie negatywnych, ang. *true negative rate* — **TNR**). Określa on liczbę predykcji prawdziwie negatywnych podzieloną przez łączną liczbę wartości negatywnych w całym zbiorze. Idealny klasyfikator uzyskuje TPR i TNR na poziomie 100%.

Niestety, większość klasyfikatorów jest daleka od doskonałości, dlatego ich skuteczność trzeba oceniać na podstawie błędów. Współczynnik predykcji fałszywie pozytywnych (ang. *false positive rate* — **FPR**) to 1 minus TNR, natomiast współczynnik predykcji fałszywie negatywnych (ang. *false negatywnych rate* — **FNR**) to 1 minus TPR. Są to tak zwane *błędy pierwszego rodzaju* i *błędy drugiego rodzaju*. W tabeli 1.2 przedstawione są relacje między opisanymi miarami.

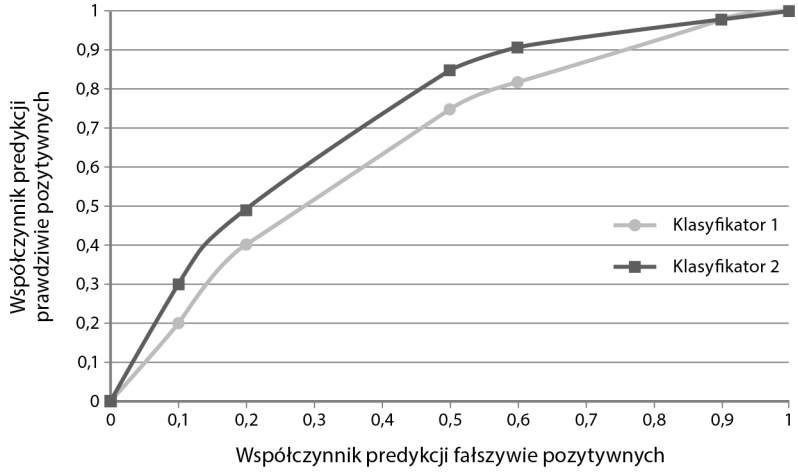
Tabela 1.2. Miary skuteczności używane do oceny inteligentnych algorytmów

Miara	Obliczenia
Współczynnik predykcji prawdziwie pozytywnych (TPR)	Predykcje prawdziwie pozytywne / rzeczywiste wyniki pozytywne
Współczynnik predykcji prawdziwie negatywnych (TNR)	Predykcje prawdziwie negatywne / rzeczywiste wyniki negatywne
Współczynnik predykcji fałszywie pozytywnych (FPR)	1 – współczynnik predykcji prawdziwie negatywnych
Współczynnik predykcji fałszywie negatywnych (FNR)	1 – współczynnik predykcji prawdziwie pozytywnych

Aby utrwalić te informacje, zastosuj miary z tabeli 1.2 do zbioru danych z tabeli 1.1. Jeśli starannie zastosujesz definicje, stwierdzisz, że TPR i TNR są równe 1/2, z czego wynika, że wartości FPR i FNR są takie same.

Załóżmy teraz, że algorytm zawiera wewnętrzny mechanizm dostosowujący jego działania — na przykład „pokrętło" dostosowujące czułość algorytmu. Gdy pokrętło jest ustawione na zero, algorytm ma bardzo niską czułość i klasyfikuje wszystkie przypadki jako fałszywe (TNR = 1, TPR = 0). Natomiast po ustawieniu pokrętła na 11 algorytm uznaje wszystkie przypadki za prawdziwe (TNR = 0, TPR = 1). Oczywiście żadna z tych sytuacji nie jest pożądana. Optymalny jest klasyfikator, który generuje wynik pozytywny tylko dla pozytywnych elementów i wynik negatywny tylko dla negatywnych elementów (TNR = 1, TPR = 1). Taki idealny mechanizm jest możliwy wyłącznie dla skrajnie uproszczonych problemów. Możesz jednak zobaczyć, jak rozwiązanie działa dla różnych ustawień pokrętła, i wykorzystać wyniki do oceny algorytmu (zobacz rysunek 1.6).

Na rysunku 1.6 pokazane są krzywe **ROC** (ang. *receiver operating characteristic*) dla dwóch fikcyjnych klasyfikatorów. Te krzywe ilustrują miary TPR i FPR dla różnych ustawień wyimaginowanego pokrętła, które zmienia parametry klasyfikatorów. Wcześniej wspomnieliśmy, że dla idealnego klasyfikatora TPR = 0 i TNR = 0 (FPR = 0). Dlatego modele zbliżone do lewego górnego rogu wykresu są teoretycznie lepsze, ponieważ lepiej radzą sobie z wyodrębnianiem klas pozytywnej i negatywnej. Na pokazanym wykresie klasyfikator 2. jest skuteczniejszy i należy go wybrać zamiast klasyfi-



Rysunek 1.6. Krzywe ROC dwóch fikcyjnych klasyfikatorów. Po zmodyfikowaniu parametru algorytmu widoczna jest zmiana w jego działaniu dla określonego zbioru danych. Im bliżej krzywa znajduje się lewego górnego rogu, tym klasyfikator jest bliższy ideałowi, ponieważ TPR = 1 i TNR = 1 (TNR = 1 – FPR)

katora 1. Inny sposób oceny skuteczności to obliczenie powierzchni pod krzywą ROC (jest to powierzchnia **AUC** — od ang. *area under the curve*, czyli powierzchnia pod krzywą). Im większa jest powierzchnia AUC, tym wyższa skuteczność modelu.

Do tej pory wszystko wygląda sensownie. Wiedz jednak, że w rozwiązaniach z obszaru analityki predykcyjnej sytuacja nie zawsze jest tak prosta. Pomyśl na przykład o ocenach kredytowych. Gdy wydasz ocenę, nie zawsze możesz prześledzić, jak dany klient będzie się zachowywał w przyszłości. Ponadto w takich scenariuszach nie próbujesz udzielić odpowiedzi; celem jest określenie wartości skorelowanej z docelową zmienną (na przykład z wiarygodnością kredytobiorcy). Choć w takich sytuacjach można posłużyć się krzywą ROC, czasem trzeba pokonać kilka dodatkowych przeszkód.

1.7. Ważne uwagi na temat inteligentnych algorytmów

Do tej pory omówiliśmy już wiele wprowadzającego materiału. Na tym etapie powinieneś dobrze (choć na ogólnym poziomie) rozumieć inteligentne algorytmy i wiedzieć, jak się nimi posługiwać. Zapewne jesteś niecierpliwy i zmotywowany, aby przejść do szczegółów. Nie zawiedziemy Cię. Każdy następny rozdział jest bogaty w nowy i wartościowy kod. Jednak zanim rozpoczniesz podróż po ekscytującym i (dla bardziej cynicznych programistów) atrakcyjnym finansowo świecie inteligentnych aplikacji, powinieneś zapoznać się z listą przydatnych informacji. Liczne z nich zapożyczyliśmy z doskonałej i przystępnej pracy Pedro Domingosa[18]. Te informacje będą pomocne w czasie lektury tej książki, a także później, w trakcie pracy w dziedzinie inteligentnych algorytmów.

[18] Pedro Domingos, *A Few Useful Things to Know About Machine Learning*, „Communications of the ACM" 55, nr 10 (2012): 78 – 87.

1.7.1. Dane nie są wiarygodne

Dane z wielu powodów mogą być niewiarygodne. To dlatego zawsze powinieneś sprawdzać, czy używane dane są godne zaufania. Dopiero potem możesz zacząć zastanawiać się nad rozwiązaniem problemu za pomocą inteligentnych algorytmów. Nawet inteligentni ludzie dochodzą zwykle do błędnych wniosków, jeśli posługują się nieprawidłowymi danymi. Poniżej znajduje się przydatna, choć niepełna lista źródeł problemów z danymi:

- Dane dostępne w trakcie rozwijania rozwiązania mogą być niereprezentatywne dla danych ze środowiska produkcyjnego. Załóżmy, że chcesz dzielić użytkowników sieci społecznościowej według wzrostu na grupy: „wysocy", „przeciętni" i „niscy". Jeśli najniższa osoba w Twoim zespole programistycznym ma 184 centymetry wzrostu, ryzykujesz tym, że nazwiesz kogoś niskim, ponieważ mierzy „tylko" 184 centymetry.
- W danych mogą występować braki. W rzeczywistości, jeśli dane nie są sztucznie generowane, prawie na pewno będą niepełne. Obsługa braku wartości to skomplikowane zadanie. Zwykle albo pozostawia się lukę w wartościach, albo zapełnia ją domyślnymi lub obliczonymi wartościami. Oba podejścia mogą prowadzić do niestabilnych rozwiązań.
- Dane mogą się zmieniać. Ktoś może zmodyfikować schemat bazy danych lub zmienić znaczenie przechowywanych w niej danych.
- Dane mogą być nieznormalizowane. Załóżmy, że analizujesz wagę grupy osób. Aby można było dojść do znaczących wniosków na podstawie wagi, jednostki miary dla wszystkich osób powinny być takie same. W całym zbiorze trzeba używać albo funtów, albo kilogramów, a nie jednej jednostki dla części osób i drugiej dla pozostałych.
- Dane mogą być niedostosowane do algorytmicznego podejścia, jakie planujesz zastosować. Dane przyjmują różne postacie i formy, którym odpowiadają *typy danych*. Niektóre zbiory danych są liczbowe, inne nie. Niektóre zbiory danych można porządkować, inne tego nie umożliwiają. Niektóre liczbowe zbiory danych są nieciągłe (na przykład liczba osób w pomieszczeniu), natomiast inne — ciągłe (na przykład temperatura lub ciśnienie atmosferyczne).

1.7.2. Wnioskowanie wymaga czasu

Obliczanie rozwiązania zajmuje czas, a szybkość reagowania aplikacji może być krytyczna dla odniesienia przez firmę finansowego sukcesu. Nie powinieneś przyjmować, że wszystkie algorytmy dla wszystkich zbiorów danych będą działały na tyle szybko, by aplikacja mogła udzielić odpowiedzi w ustalonym limicie czasu. Należy przetestować szybkość algorytmu z uwzględnieniem charakterystyki działania aplikacji.

1.7.3. Wielkość ma znaczenie!

W kontekście inteligentnych aplikacji wielkość ma znaczenie! Wielkość danych jest istotna w dwóch aspektach. Pierwszy związany jest ze wspomnianym wcześniej czasem reagowania. Drugi dotyczy możliwości uzyskania wartościowych wyników dla dużych

zbiorów danych. Możliwe, że aplikacja potrafi generować doskonałe rekomendacje filmów lub muzyki dla około 100 użytkowników, ale dla grup około 100 000 osób zwraca mało przydatne wyniki.

Ponadto, co związane jest z „przekleństwem wymiarów" (zobacz rozdział 2.), przekazanie większej ilości danych do prostego algorytmu często daje znacznie lepsze wyniki niż budowanie bardziej skomplikowanego klasyfikatora. Gdy przyjrzysz się dużym korporacjom (takim jak Google), które wykorzystują bardzo duże ilości danych, powinieneś docenić zarówno umiejętność obsługi dużych zbiorów danych treningowych, jak i złożoność oraz zaawansowanie rozwiązań klasyfikujących.

1.7.4. Różne algorytmy skalują się w odmienny sposób

Nie zakładaj, że możliwe jest skalowanie inteligentnej aplikacji przez proste dodanie kolejnych maszyn. W ogóle nie powinieneś przyjmować, że rozwiązanie jest skalowalne. Niektóre algorytmy się skalują, natomiast inne nie. Załóżmy, że wśród miliardów tytułów chcesz znaleźć grupy artykułów informacyjnych z podobnymi nagłówkami. Nie wszystkie algorytmy klastrowania mogą działać równolegle. Powinieneś uwzględnić skalowalność na etapie projektowania aplikacji. W niektórych sytuacjach możliwy jest podział danych i zastosowanie inteligentnego algorytmu równolegle do mniejszych zbiorów danych. Algorytmy wybrane w trakcie projektowania mają czasem wersje równoległe (współbieżne), jednak powinieneś sprawdzić to już na początku prac, ponieważ na podstawie użytych algorytmów tworzona jest rozbudowana infrastruktura i logika biznesowa.

1.7.5. Nie wszystko jest gwoździem!

Możliwe, że zetknąłeś się już z twierdzeniem: „Jeśli masz tylko młotek, wszystko wygląda jak gwóźdź". Oznacza to, że nie da się za pomocą tego samego algorytmu rozwiązać wszystkich problemów wymagających inteligentnych aplikacji.

Inteligentne aplikacje są jak każde inne oprogramowanie — mają określony obszar zastosowań i pewne ograniczenia. Koniecznie starannie przetestuj swoje ulubione rozwiązanie w nowych obszarach. Ponadto zalecamy, aby każdy problem analizować z nowej perspektywy. Inne algorytmy mogą rozwiązywać określone problemy w wydajniejszy lub bardziej dogodny sposób.

1.7.6. Dane to nie wszystko

Algorytmy uczenia maszynowego nie działają w magiczny sposób i wymagają indukcji, aby wyjść poza dane treningowe i móc przetwarzać nieznane wcześniej informacje. Jeśli masz już bogatą wiedzę na temat zależności w danych, dobrą reprezentacją mogą być modele graficzne, pozwalające łatwo przedstawić uprzednią wiedzę[19]. Staranne przemyślenie tego, co już wiadomo w danej dziedzinie (z uwzględnieniem danych), pomaga budować skuteczne klasyfikatory.

[19] Judea Pearl, *Probabilistic Reasoning in Intelligent Systems* (Morgan Kaufmann Publishers, 1988).

1.7.7. Czas treningu może się zmieniać

W niektórych zastosowaniach czas generowania rozwiązania może się znacznie zmieniać przy tylko niewielkiej zmianie parametrów. Zwykle użytkownicy oczekują, że po zmianie parametrów problemu nadal będzie można go rozwiązać w tym samym czasie. Gdy używasz metody zwracającej odległość między dwoma lokalizacjami geograficznymi na Ziemi, spodziewasz się, że uzyskasz wynik w tym samym czasie niezależnie od uwzględnianych lokalizacji. Jednak nie we wszystkich problemach jest to prawdą. Pozornie niewinna zmiana w danych może prowadzić do znacznej zmiany czasu generowania wyniku. Bywa, że ten czas zmienia się z sekund na godziny!

1.7.8. Celem jest generalizacja

Jedną z pułapek, w którą praktycy z obszaru uczenia maszynowego wpadają najczęściej, jest koncentracja na procesie pracy i zapomnienie o ostatecznym celu, a jest nim generalizacja analizowanego zjawiska. W fazie testów niezbędne jest stosowanie metod, które pozwalają ocenić ogólność rozwiązania (pomijanie danych testowych w czasie treningu, walidacja krzyżowa itd.). Nic jednak nie zastąpi używania od początku odpowiedniego zbioru danych! Jeśli próbujesz uogólnić proces o milionie atrybutów za pomocą kilkuset przykładów testowych, skupianie się na osiągnięciu wysokiej trafności nie ma żadnego sensu.

1.7.9. Ludzka intuicja nie zawsze się sprawdza

Gdy przestrzeń cech rośnie, następuje eksplozja kombinatoryczna, jeśli chodzi o liczbę możliwych wartości wejściowych. Dlatego już dla umiarkowanie rozbudowanego zbioru cech możesz zobaczyć tylko bardzo niewielki ułamek wszystkich możliwych danych wejściowych. Bardziej problematyczne jest to, że wraz z rosnącą liczbą cech ludzka intuicja zaczyna zawodzić. Na przykład większość masy w wielowymiarowym rozkładzie normalnym znajduje się nie blisko średniej, ale w „otoczce" wokół niej[20]. Budowanie prostego klasyfikatora dla niewielkiej liczby wymiarów jest łatwe, jednak przy większej liczbie wymiarów trudno jest zrozumieć zależności w danych.

1.7.10. Pomyśl o zaprojektowaniu nowych cech

Prawdopodobnie zatknąłeś się ze stwierdzeniem: „śmieci na wejściu, śmieci na wyjściu". Ma ono duże znaczenie w trakcie budowania rozwiązań z obszaru uczenia maszynowego. Bardzo istotne jest tu zrozumienie dziedziny problemowej, a ustalenie zestawu cech uwidaczniającego badane zjawisko może mieć duży wpływ na trafność i ogólność klasyfikatora. Nie wystarczy przekazać klasyfikatorowi wszystkich dostępnych danych i liczyć na cud.

1.7.11. Poznaj wiele różnych modeli

Coraz popularniejsze stają się zestawy modeli, ponieważ pozwalają ograniczyć zmienność w procesie klasyfikacji kosztem tylko niewielkiego błędu systematycznego. W konkursie Netflix Prize pierwsze i drugie miejsce zajęły zestawy warstwowe (w których dane

[20] Domingos, *A Few Useful Things to Know About Machine Learning*.

wyjściowe każdego klasyfikatora są przekazywane do klasyfikatora wyższego poziomu) obejmujące ponad 100 jednostek uczących się. Wiele osób uważa, że dzięki takim technikom w przyszłości klasyfikatory będą skuteczniejsze, jednak to podejście powoduje utworzenie nowej warstwy pośredniej, którą osoba niebędąca ekspertem musi poznać, aby zrozumieć funkcjonowanie systemu.

1.7.12. Korelacja nie oznacza związku przyczynowo-skutkowego

Warto przypomnieć tę często przytaczaną kwestię. Można też zilustrować ją za pomocą żartobliwego eksperymentu myślowego: „Globalne ocieplenie, trzęsienia ziemi, huragany i inne katastrofy naturalne są bezpośrednim efektem zmniejszania się liczby piratów od XIX wieku”[21]. To, że dwie zmienne są skorelowane, nie oznacza ich połączenia związkiem przyczynowo-skutkowym. Często istnieje trzecia (a nawet czwarta lub piąta!) nieobserwowalna zmienna, która wpływa na pozostałe. Korelację należy traktować jak oznakę możliwej przyczynowości zasługującą na dalsze analizy.

1.8. Podsumowanie

- Przedstawiliśmy tu bardzo ogólny obraz inteligentnych algorytmów i zaprezentowaliśmy wiele przykładowych rozwiązań używanych w praktyce.
- Inteligentny algorytm cechuje się tym, że potrafi modyfikować swoje działanie na podstawie otrzymywanych danych.
- Pokazaliśmy kilka antywzorców projektowych dotyczących inteligentnych algorytmów. Mamy nadzieję, że posłużą one jako ostrzeżenie dla praktyków z tej dziedziny.
- Wyjaśniliśmy, że inteligentne algorytmy można ogólnie podzielić na trzy kategorie: sztuczna inteligencja, uczenie maszynowe i analityka predykcyjna. W tej książce najwięcej miejsca poświęcamy dwóm ostatnim z tych klas. Dlatego jeśli tylko pobieżnie przejrzałeś poświęcone im fragmenty, powinieneś ponownie się z nimi zapoznać, aby zapewnić sobie solidną podstawową wiedzę.
- Przedstawiliśmy wybrane podstawowe miary takie jak krzywa ROC. Obszar pod krzywą ROC często jest używany do oceny względnej skuteczności modeli. Pamiętaj jednak, że istnieje wiele sposobów oceny skuteczności. Tu pokazaliśmy tylko podstawowe z nich.
- Zaprezentowaliśmy też przydatne informacje zdobyte przez społeczność pracującą nad inteligentnymi algorytmami. Będą one nieocenionym kompasem w trakcie podróży po tym obszarze.

[21] Bobby Henderson, *An Open Letter to Kansas School Board*, Verganza (1 lipca 2005), *http://www.venganza.org/about/open-letter*.

Skorowidz

A

adekwatność testów A/B, 191
agent, 133
AI, artificial intelligence, 26
aktywowanie sieci neuronowej, 175
algorytm
  adaptacyjnej kwantyzacji wektorowej, 103
  GMM, 40, 59
  k najbliższych sąsiadów, 103
  k-średnich, 45, 49, 59
  Lloyda, 45, 49
  uczenia perceptronów, 160
algorytmy klasyfikacji, 102, 104
analityka predykcyjna, PA, 26, 29
analiza głównych składowych, PCA, 41, 61, 63
aplikacja Google Now, 21

B

bandyta kontekstowy, 209
biblioteka
  SVDLIBC, 86
  Vowpal Wabbit, 138
  VW, 143
błąd, 171
  systematyczny, 41, 65
  średniokwadratowy, 93, 95
BRBM, Bernoulli Restricted Boltzmann Machines, 176

C

cechy, 41, 137
  ciągłe, 114
  kategorialne, 112
  macierzy kwadratowych, 61
centroid, 49
CF, collaborative filtering, 76
CPC, cost per click, 136, 220
CPI, cost per impression, 136
CPM, 136, 220
CTR, 136
cykl życia
  inteligentnych algorytmów, 23
  klasyfikatora, 105

D

dane, 41
  o kliknięciach, 234
  wyjściowe, 145
decyzje, 134, 150, 186
dekompozycja zbioru danych, 64
dokonywanie wyboru, 185
dopasowanie, 110
  plików cookie, 130, 131
drzewo GDBT, 94
DSP, demand-side platform, 129
działanie perceptronu, 162

E

elementy struktury odniesienia, 100
EM, expectation maximization, 40

F

filtrowanie kolaboratywne, CF, 68, 76
  według użytkowników, 77
FNR, false negatywnych rate, 32
FPR, false positive rate, 32
funkcja gęstości prawdopodobieństwa, 200–202
funkcje
  aktywacji, 168
  odległości euklidesowej, 74

G

gaussowski model mieszany, GGM, 40, 52, 55
generowanie
  krzywej ROC, 147
  rekomendacji, 85
  sztucznych danych, 181
GGM, Gaussian mixture model, 40
giełda, 130
GLM, generalized linear models, 169
GMM, 52

H

HDFS, Hadoop Distributed File System, 241
HFT, high-frequency trading, 129
histogram rozkładu Gaussa, 53

I

implementowanie wykrywania oszustw, 111
importowanie
  bibliotek, 112
  zbioru danych, 112
informacje
  o kontekście, 135
  o przestrzeni reklamowej, 135
  o użytkowniku, 135
inteligencja, 30
inteligentna sieć, 19
inteligentny algorytm, 21–24, 33
  cykl życia, 23
  generalizacja, 36
  Google Now, 21
  klasy, 26
  ludzka intuicja, 36
  miary skuteczności, 32
  ocena działania, 30
  skalowanie, 35
  wiarygodność danych, 34
  wielkość, 34
  wnioskowanie, 34
intuicja, 36
inżynieria cech, 137

J

jednostki ukryte, 183

K

kalibrowanie modelu, 148
KISS, 26
klasa
  RatingCountMatrix, 81
  SimilarityMatrix, 80
klastrowanie, 43
  zbioru danych, 51, 57
klastry, 46
klasy inteligentnych algorytmów, 26
klasyfikator, 105
klasyfikowanie, 97, 122
kodowanie
  cech kategorialnych, 113
  one-hot, 114
konkurs Netflix Prize, 93
korelacja, 37
koszt
  kliknięcia, CPC, 136
  wyświetlenia reklamy, CPI, 136
krzywa ROC, 32, 147
k-średnie, 45, 49

L

liczba cech, 51
lokalizacja, 22

Ł

łączenie systemów, 238, 240

M

macierz
  błędów, 120
  kowariancji, 57, 62
  kwadratowa, 61
maksymalizacja wartości oczekiwanej, EM, 40, 49, 55, 65
maszyny
  BRBM, 177
  RBM, 180
metoda
  Lanczosa, 86
  make_ellipses, 60
  recommend, 78
metody dywergencji kontrastowej, 180

miara
  pierwiastka błędu średniokwadratowego, 95
  RMSE, 95
  skuteczności, 32
mieszanie, 142
ML, machine learning, 26
model
  Bernoulliego, 176
  liniowy, 169
  MCP, 157
modelowanie wiedzy, 196
monitorowanie reklam, 132

N

narzędzie VW, 138, 145
nierówność trójkąta, 72

O

obiekt typu SimpleProducer, 229
obliczanie
  miary RMSE, 96
  podobieństwa, 70
  różnic, 107
ocenianie
  działania inteligentnych algorytmów, 30, 32
  predykcji, 31
  systemu rekomendacji, 94
oczekiwany poziom straty, 203, 207
oddzielanie cech kategorialnych, 113
odległość, 69
  euklidesowa, 73
oferty, 131
ograniczona maszyna Boltzmanna, 176
osie danych, 60

P

PA, predictive analytics, 26
pakiet
  PyBrain, 172, 183
  scikit-learn, 41, 42, 147
PCA, principal component analysis, 41
perceptron, 156, 162, 169
  dwuwarstwowy, 166
  wielowarstwowy, 164, 174
platforma DSP, 129, 131
plik server.properties, 227
pliki cookie, 130
płaskie struktury odniesienia, 99
pobieranie danych, 219
podejmowanie decyzji, 134, 136, 150, 186
podobieństwo, 69, 70, 73, 75
  prawdopodobieństw, 205
potwierdzanie przesłania komunikatów, 229
powiadomienie o wygranej, 132
prawdopodobieństwo wystąpienia zdarzenia, 110
predykcja, 31
  kliknięć, 138
predyktor prawdopodobieństwa, 109
problem
  bandytów, 210
  klasyfikacji binarnej, 120
  wielorękiego bandyty, 192
prognozowanie
  kliknięć, 127
  ocen, 79
  zdarzeń, 150
prognozowany współczynnik kliknięć, 136
propagacja wsteczna, 167–170
przegląd klasyfikatorów, 101
przekleństwo wymiarów, 44
przestrzeń cech, 49, 65
przesyłanie dzienników, 222
przygotowywanie danych, 135, 141
publikowanie komunikatów, 224

R

regresja
  liniowa, 106
  logistyczna, 106, 117, 143, 169, 181
reguła KISS, 26
rejestrowanie danych, 221
reklama, 132
  internetowa, 220
rekomendacje, 67, 76, 84
  filtrowanie kolaboratywne, 77
  generowanie, 85
  oparte na modelu, 82, 87
  rozkład SVD, 82, 84, 90
replikacja, 226, 230
reprezentacja ukrytych jednostek, 182
RMSE, root-mean square error, 95
ROC, receiver operating characteristic, 32
rozkład
  a priori, 199
  beta, 198
  Gaussa, 52, 55
  macierzy na czynniki, 85
  prawdopodobieństwa, 56, 202
  według wartości osobliwych, SVD, 68, 82–85, 90
równoważenie, 142

S

SGD, stochastic gradient descent, 142
sieci neuronowe, 155
  perceptron, 156
  RBM, 179
  trening, 158
  wielowarstwowe, 172
skala prawdopodobieństwa, 205
skalowanie inteligentnej aplikacji, 35
skumulowany oczekiwany poziom straty, 206
spadek wzdłuż gradientu, SGD, 142
statystyczne algorytmy klasyfikacji, 104
stosowanie strategii bayesowskiej, 198
strategia
  bayesowska, 195, 197, 204, 207
  najpierw epsilon, 193
  zachłanna, 194
  zmniejszania epsilonu, 195
struktura, 41
strukturalne algorytmy klasyfikacji, 102
subskrypcja, 226
SVD, singular value decomposition, 68, 82–85, 90
SVM, support vector machines, 28
symetria, 72
synchronizowane repliki, 230
system
  Hadoop, 240
  HDFS, 241
  Kafka, 224
    klaster, 234
    poziomy potwierdzeń, 230
    publikowanie, 228
    publikowanie komunikatów, 224
    rejestrowanie danych, 236
    replikacja, 226, 231
    subskrypcja, 226
    wzorce projektowe, 238
  Storm, 238
systemy
  podejmowania decyzji, 134, 150
  rekomendacji, 69, 76
sztuczna inteligencja, AI, 26
szum, 41

T

teoria propagacji wstecznej, 170
testowanie modelu, 146
testy A/B, 187, 207
TNR, negative rate, 32
TPR, positive rate, 31
transformacje osi danych, 60
trening
  modelu, 137, 143
  perceptronu, 160
  sieci neuronowej, 173
  z wykorzystaniem propagacji wstecznej, 167
treningowy zbiór danych, 142
tworzenie
  wielowarstwowego perceptronu, 172
  zbioru danych, 180

U

uczenie
  głębokie, 153, 175
  maszynowe, ML, 26, 28
  oparte na energii, 179
umieszczanie reklamy, 132
uogólnione modele liniowe, GLM, 169

V

VW, Vowpal Wabbit, 138
  format danych, 138

W

wagi sieci neuronowej, 175
wartość
  oczekiwana, 55
  własna, 61
wektor własny, 61
wieloręki bandyta, 192, 208
wielowarstwowa sieć jednokierunkowa, 164
  neuronowa, 172
wizualizacja sieci głębokiej, 155
współczynnik
  kliknięć, CTR, 136
  konwersji, 191
  predykcji, 32
wykrywanie oszustw, 106, 111
wymagania stawiane agentowi, 133
wyniki, 119
wzór
  na błąd, 171
  na podobieństwo, 75

Z

zarządzanie zbieraniem danych, 222
zastosowania inteligentnej sieci
  autonomiczne samochody, 215
  internet rzeczy, 214
  opieka zdrowotna, 215
  sieć semantyczna, 216
  spersonalizowane fizyczne reklamy, 216
zbiór danych, 141, 142, 180
  Iris, 41, 42
  MovieLens, 69
zmienne kategorialne, 117

Określenie „inteligentna sieć” może przywodzić na myśl futurystyczną wizję maszyn przejmujących kontrolę nad światem i niszczących ludzkość, jednak w rzeczywistości jest związane z rozwojem technologii. Odnosi się do oprogramowania, które potrafi się uczyć i reagować na zachowania użytkowników. Oznacza też projektowanie i implementację inteligencji maszynowej. Inteligentna sieć rozwija się tu i teraz — znajomość zagadnień uczenia maszynowego i budowy inteligentnych algorytmów jest bardzo potrzebna inżynierom oprogramowania!

Niniejsza książka jest przeznaczona dla osób, które chcą projektować inteligentne algorytmy, a przy tym mają podstawy z zakresu programowania, matematyki i statystyki. Przedstawiono tu schematy projektowe i praktyczne przykłady rozwiązań. Opisano algorytmy, które przetwarzają strumienie danych pochodzące z internetu, a także systemy rekomendacji i klasyfikowania danych za pomocą algorytmów statystycznych, sieci neuronowych i uczenia głębokiego. Mimo że przyswojenie tych zagadnień wymaga wysiłku, bardzo ułatwi implementację nowoczesnych, inteligentnych aplikacji!

W tej książce między innymi:

- wprowadzenie do problemów algorytmów inteligentnych
- systemy rekomendacji i filtrowanie kolaboratywne
- wykorzystanie regresji logistycznej do wykrywania oszustw
- uczenie głębokie, uczenie na żywo i renesans sieci neuronowych
- podejmowanie decyzji
- perspektywy inteligentnej sieci

Dr Douglas McIlwraith jest ekspertem w dziedzinie uczenia maszynowego. Zajmuje się analizą danych w londyńskiej agencji reklamowej. Prowadził badania w dziedzinach systemów rozproszonych, robotyki i zabezpieczeń.

Dr Haralambos Marmanis jest pionierem w obszarze technik uczenia maszynowego w rozwiązaniach przemysłowych. Od 25 lat rozwija profesjonalne oprogramowanie.

Dmitry Babenko projektuje złożone systemy dla firm z takich branż jak bankowość, ubezpieczenia, zarządzanie łańcuchem dostaw i analityka biznesowa.

Inteligentny algorytm wyławia perły w strumieniach danych!

Helion

księgarnia internetowa
http://helion.pl

zamówienia telefoniczne
0 801 339900
0 601 339900

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
http://helion.pl

Sprawdź najnowsze promocje:
http://helion.pl/promocje
Książki najchętniej czytane:
http://helion.pl/bestsellery
Zamów informacje o nowościach:
http://helion.pl/nowosci

ISBN 978-83-283-3250-8

Informatyka w najlepszym wydaniu

cena: 54,90 zł