

Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH



Instytut Instytut Mikroelektroniki i Optoelektroniki

Praca dyplomowa inżynierska

na kierunku Elektronika i Telekomunikacja
w specjalności Inżynieria Komputerowa

Zastosowanie wybranych algorytmów uczenia maszynowego w
zarządzaniu łańcuchem dostaw

Jarosław Wanczewski

Numer albumu 309470

promotor
dr inż. Marek Niewiński

WARSZAWA 2023

Zastosowanie wybranych algorytmów uczenia maszynowego w zarządzaniu łańcuchem dostaw

Abstract.

To jest moje streszczenie. Moja praca dotyczy...



.....
miejscowość i data

.....
imię i nazwisko studenta

.....
numer albumu

.....
kierunek studiów

OŚWIADCZENIE

Świadomy/-a odpowiedzialności karnej za składanie fałszywych zeznań oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie, pod opieką kierującego pracą dyplomową.

Jednocześnie oświadczam, że:

- niniejsza praca dyplomowa nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.) oraz dóbr osobistych chronionych prawem cywilnym,
- niniejsza praca dyplomowa nie zawiera danych i informacji, które uzyskałem/-am w sposób niedozwolony,
- niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadawaniem dyplomów lub tytułów zawodowych,
- wszystkie informacje umieszczone w niniejszej pracy, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami,
- znam regulacje prawne Politechniki Warszawskiej w sprawie zarządzania prawami autorskimi i prawami pokrewnymi, prawami własności przemysłowej oraz zasadami komercjalizacji.

Oświadczam, że treść pracy dyplomowej w wersji drukowanej, treść pracy dyplomowej zawartej na nośniku elektronicznym (płycie kompaktowej) oraz treść pracy dyplomowej w module APD systemu USOS są identyczne.

.....
czytelny podpis studenta

Spis treści

1. Wstęp	9
1.1. Tło	9
1.2. Opis problemu	9
1.3. Cele	10
1.4. Zakres badania	10
2. Przegląd	11
2.1. Zarządzanie łańcuchem dostaw	11
2.1.1. Teoria Zarządzania Łańcuchem Dostaw	11
2.1.2. Cele Zarządzania Łańcuchem Dostaw	13
2.1.3. Znaczenie zarządzania Łańcuchem Dostaw	13
2.1.4. Składniki Efektywnego Zarządzania Łańcuchem Dostaw	15
2.2. Maszynowe Uczenie	16
2.2.1. Sztuczna Inteligencja (SI)	16
2.2.2. Maszynowe Uczenie	18
2.2.3. Zastosowania Maszynowego Uczenia	18
2.2.4. Rodzaje Maszynowego Uczenia	18
2.3. Uczenie maszynowe w zarządzaniu łańcuchem dostaw	19
2.3.1. Prognozowanie popytu	19
2.3.2. Optymalizacja zapasów	31
2.3.3. Planowanie produkcji	35
2.4. Kluczowe algorytmy uczenia maszynowego dla łańcucha dostaw	46
2.4.1. Uczenie nienadzorowane (Unsupervised Learning)	50
2.4.2. Uczenie nadzorowane (supervised Learning)	71
3. Metodologia	72
3.1. Gromadzenie i przygotowanie danych	72
3.2. Wybór algorytmu uczenia maszynowego:	72
3.3. Rozwój modelu	72
3.4. Metryki oceny modelu	72
4. Wybrane algorytmy uczenia maszynowego	73
4.1. Regresja liniowa	73
4.2. Drzewa decyzyjne	73
4.3. Sieci neuronowe	73
4.4. Maszyny wektorów nośnych	73
4.5. Algorytmy klastrowania	73
5. Studium przypadku	74
5.1. Studium przypadku 1: Optymalizacja zapasów	74
5.2. Studium przypadku 2: Prognozowanie popytu	74

5.3. Studium przypadku 3: Wybór dostawcy	74
5.4. Studium przypadku 4: Planowanie produkcji	74
6. Wyniki i dyskusja	75
6.1. Porównanie wydajności algorytmów	75
6.2. Interpretacja wyników	75
6.3. Implikacje dla zarządzania łańcuchem dostaw	75
7. Wyzwania i przyszłe kierunki	76
7.1. Wyzwania we wdrażaniu ML w SCM	76
7.2. Przyszłe kierunki badań	76
7.3. Względy etyczne i dotyczące prywatności	76
8. Wniosek	77
8.1. Podsumowanie ustaleń	77
8.2. Wkład	77
8.3. Zastosowania praktyczne	77
8.4. Wnioski i uwagi końcowe	77
Bibliografia	79
Wykaz symboli i skrótów	85
Spis rysunków	85
Spis tabel	85
Spis załączników	85

1. Wstęp

1.1. Tło

Współczesny świat (postępująca globalizacja) charakteryzuje się dynamicznymi zmianami, rosnącą konkurencją oraz rosnącym zapotrzebowaniem na coraz bardzo skomplikowane usługi i produkty, stawia to przed przedsiębiorstwami wiele wyzwań związanych z efektywnym zarządzaniem łańcuchem dostaw.

W szczególności kluczowe technologie mają ogromne znaczenie w długoterminowej perspektywie konkurencyjności firm. Dlatego warto rozważyć ich zastosowanie. Współczesne kluczowe technologie XXI wieku znane są jako sztuczna inteligencja (AI). Analogicznie do ludzkiego poznania, termin "sztuczna inteligencja" obejmuje systemy zdolne do przetwarzania informacji, rozumienia języka, wykonywania działań i skoncentrowania się na celach, a także zdobywania wiedzy do rozwiązywania problemów, przy wykorzystaniu uczenia maszynowego. Aby wykorzystać tę zdolność uczenia się przez system, uczenie maszynowe (ML) wyłoniło się jako niezależna dziedzina badań i technologii w kontekście sztucznej inteligencji. W odróżnieniu od ręcznego kodowania pojedynczych rozwiązań, takich jak korzystanie z reguł czy ontologii, wiedza z zakresu uczenia maszynowego jest automatycznie dostarczana przez odpowiednie systemy, które wykorzystują algorytmy oparte na empirycznych danych do rozwiązywania problemów.[1]

Sztuczna inteligencja jest obecnie jedna z najszybciej rozwijających się technik, mającą w sobie potencjał do przełomowego wpływu na sposób organizacji i funkcjonowania zarówno społeczeństwa jako całości, jak i pojedynczych osób. Już obecnie widzimy oddziaływanie SI na gospodarkę, w szczególności na produkcję przemysłową, gdzie tzw. przemysł 4.0, oparty na szerokiej robotyzacji zastępuje tradycyjne formy wytwarzania produktów.[2]

1.2. Opis problemu

Dawniej począwszy od kadry kierowniczej wyższego szczebla po kadrę kierowniczą pierwszej linii, mierząc się ze złożonymi i krytycznymi decyzjami to tradycyjnie takie decyzje zależały od ich doświadczenia i osądu. Jednakże rynek przestawił się na krótkie serie produkcyjne, aby zaspokoić szybko zmieniający się popyt, a koszty zostały obniżone na rzecz metod produkcji „dokładnie na czas”. decyzje stały się bardziej złożone. Jednocześnie procesy operacyjne (produkcja) staje się bardziej zautomatyzowana i zintegrowana, co umożliwi większą kontrolę nad łańcuchem dostaw. Dlatego ostatnio coraz większą uwagę poświęca się technikom sztucznej inteligencji (AI).[3]

W dobie rozwijającej się i powszechnie dostępnej technologii oraz dostępu do aktualnej informacji nie wystarczy oferować wyroby i usługi w tzw. „standardzie”. Wyróżniać się wśród konkurencji, to znaczy działać nieszablonowo, oferować wyroby i usługi dedykowane, „szyte na miarę” pod indywidualne potrzeby klientów, przy zachowaniu wymaganego poziomu jakości. [4]

Łańcuch dostaw to obszar istotny, który ma wpływ na konkurencyjność, elastyczność i zdolność do częstego i szybkiego dostosowywania się do coraz szybciej zmieniającego się rynku. W ten obszar doskonale wkomponowują się technologie z zakresu uczenia maszynowego, które stają się znaczącym narzędziem wspierającym lub całkowicie decydującym w zarządzaniu łańcuchem dostaw. W dzisiejszym środowisku biznesowym, które cechuje się rosnącą ilością dostępnych danych oraz coraz bardziej złożonymi danymi, algorytmy uczenia maszynowego zaczynają stanowić kluczowy element umożliwiający podejmowanie bardziej poinformowanych decyzji, redukcję kosztów oraz zwiększenie efektywności operacyjnej.

Biorąc pod uwagę złożoność zadań planowania, zarządzania i kontroli w przemysłowych łańcuchach wartości, aplikacje ML uznawane są za niezwykle istotne dla wspierania i autonomicznej realizacji logistycznych procesów decyzyjnych[1]

1.3. Cele

Celem jest zbadanie i ocena potencjału zastosowania wybranych algorytmów uczenia maszynowego w zarządzaniu łańcuchem dostaw.

1.4. Zakres badania

Praca skupia się na analizie, projektowaniu oraz implementacji algorytmów uczenia maszynowego w różnych częściach zarządzania łańcuchem dostaw, takich jak prognozowanie popytu, planowanie produkcji, zarządzanie zapasami.

2. Przegląd

W rozdziale tym dokonuje przeglądu istotnych zagadnień związanych z zarządzaniem łańcuchem dostaw oraz uczeniem maszynowym. Przedstawimy definicje, teorie, rodzaje zarządzania łańcuchem dostaw i maszynowego uczenia. Następnie omówiam definicje i zastosowania maszynowego uczenia w zarządzaniu łańcuchem dostaw.

2.1. Zarządzanie łańcuchem dostaw

W niniejszym podrozdziale dokładnie omówiam najważniejsze zagadnienia związane z zarządzaniem łańcuchem dostaw (definicje, cele oraz znaczenie). Ponadto wyjaśniam z jakich elementów składa się zarządzanie łańcuchem dostaw.

2.1.1. Teoria Zarządzania Łańcuchem Dostaw

Wyjaśnienie Zarządzania Łańcuchem Dostaw

Zarządzanie łańcuchem dostaw (ang. Supply Chain Management, SCM) jest kompleksowym podejściem do planowania, kontrolowania i monitorowania wszystkich procesów związanych z dostarczaniem produktów lub usług od dostawców do klientów końcowych. Jest to obszar, który obejmuje wiele etapów, począwszy od zaopatrzenia, produkcji, dystrybucji, aż po dostarczenie produktu lub usługi do ostatecznego użytkownika.

Zarządzanie łańcuchem dostaw (ang. Supply Chain Management – SCM) – zarządzanie przepływami między ogniwami łańcuchem dostaw. Umożliwia projektowanie, planowanie, realizację, kontrolę oraz monitoring łańcucha dostaw. [5]

Wg. Encyklopedii zarządzania Łańcuch dostaw - obejmuje wszelkie czynności związane z transportem oraz przeróbką towarów, wspominając tutaj również o początkowym etapie, czyli pozyskiwaniu wszelkiego rodzaju surowców oraz etapie końcowym tj. dostarczeniu produktu konsumentom. Pojęcie to zawiera również przepływ informacji, które są istotne podczas całego procesu. [6]

Zarządzanie łańcuchem dostaw obejmuje wszystkie działania, które przekształcają surowce w gotowe produkty i oddają je w ręce klientów. Może to obejmować określanie źródła dostaw, projektowanie, produkcję, magazynowanie, wysyłkę i dystrybucję. Celem SCM jest poprawa wydajności, jakości, produktywności i zadowolenia klientów. [7]

Łańcuch dostaw jest to skoordynowana sieć wzajemnych powiązań logistyczno - operacyjnych, która obejmuje wszelkie firmy, obiekty i działania biznesowe zaangażowane w pozyskiwanie, opracowywanie, wytwarzanie i dostarczanie produktów. Każda firma tworzy swój własny łańcuch dostaw, aby móc wytwarzać produkty i wprowadzać je na rynek. Może również sama być ogniwem w łańcuchach dostaw innych firm. Działania łańcucha dostaw przekształcają zasoby naturalne, surowce i komponenty w gotowy produkt, który jest dostarczany do użytkownika końcowego. [8]

(SCM) zajmuje się systemem zakupów (zakup surowców/komponentów), zarządzaniem operacyjnym (zapewnienie produkcji wysokiej jakości produktów z dużą szybkością, dobrą elastycznością i niskimi kosztami produkcji), logistyką i kanałami marketingowymi, dzięki któremu surowce można przekształcić w gotowe produkty i dostarczyć klientom końcowym. Węższa definicja zarządzania łańcuchem dostaw to „projektowanie, planowanie, realizacja, kontrola i monitorowanie działań w łańcuchu dostaw w celu tworzenia wartości netto, budowania konkurencyjnej infrastruktury, wykorzystania światowej logistyki, synchronizacji podaży z popytem i pomiaru wydajności globalnie”. Może to obejmować przemieszczanie i przechowywanie surowców, zapasów w toku, wyrobów gotowych i kompleksową realizację zamówień od punktu pochodzenia do punktu konsumpcji. [9]

Historia

Łańcuchy dostaw istnieją od czasów starożytnych, poczynając od pierwszego wytworzonego i sprzedanego produktu. Wraz z nadejściem industrializacji zarządzanie łańcuchem dostaw stało się bardziej złożone i pozwoliło przedsiębiorstwom na efektywniejsze wytwarzanie i dostarczanie towarów i usług. Na przykład wprowadzona przez Henry'ego Forda standaryzacja części samochodowych okazała się przełomem, który pozwolił na masową produkcję towarów, aby sprostać wymaganiom rosnącej liczby klientów. Z biegiem czasu kolejne zmiany (takie jak wprowadzenie na rynek komputerów) systematycznie zwiększały poziom zaawansowania systemów SCM. Systemy te pozostawały jednak przez wiele lat zasadniczo liniową, autonomiczną funkcją zarządzaną przez specjalistów ds. łańcucha dostaw. Sytuacja zmieniła się diametralnie wraz z pojawieniem się Internetu, innowacji technologicznych i gospodarki globalnej opartej na popycie. Obecnie zarządzanie łańcuchem dostaw nie jest już funkcją liniową, ale raczej złożonym zbiorem niejednorodnych sieci dostępnych całodobowo. W centrum tych sieci znajdują się konsumenci oczekujący realizacji swoich zamówień w wybrany przez nich sposób. [10]

Pochodzenie terminu.

W 1982 roku Keith Oliver, konsultant w firmie Booz Allen Hamilton, w wywiadzie dla Financial Times wprowadził do domeny publicznej termin „zarządzanie łańcuchem dostaw”. W 1983 roku WirtschaftsWoche w Niemczech opublikowało po raz pierwszy wyniki wdrożonego tzw. „projektu zarządzania łańcuchem dostaw”, kierowanego przez Wolfganga Partscha. W połowie lat 90. termin „zarządzanie łańcuchem dostaw” zyskał na popularności, gdy ukazała się lawina artykułów i książek na ten temat. Pierwotnie łańcuchy dostaw zdefiniowano jako obejmujące wszystkie działania związane z przepływem i przetwarzaniem towarów od surowców do użytkownika końcowego lub konsumenta końcowego, a także powiązane przepływy informacji. Mentzer i in. uważają za godne odnotowania, że te wcześniejsze definicje obejmowały konsumenta końcowego. Zarządzanie łańcuchem

dostaw zostało następnie zdefiniowane jako integracja działań w łańcuchu dostaw poprzez ulepszone relacje w łańcuchu dostaw w celu osiągnięcia przewagi konkurencyjnej. [9]

2.1.2. Cele Zarządzania Łańcuchem Dostaw

Najczęściej opisywanymi celami łańcucha dostaw w ujęciu logistycznym jest:

- Minimalizacja kosztów wynikających z przepływu towarów i informacji przy zachowaniu dobrego poziomu obsługi klienta
- Krótki czas realizacji zamówień oraz bezproblemowość i elastyczność dostaw
- Optymalizacja poziomu zapasów wraz z dostosowaniem się do potrzeb rynku [6]

Podsumowując: Głównym celem zarządzania łańcuchem dostaw jest zapewnienie, że produkty lub usługi są dostarczane w odpowiednim czasie, w optymalnych ilościach, przy minimalnych kosztach oraz z zachowaniem wysokiej jakości. SCM dąży do zintegrowania wszystkich elementów łańcucha dostaw, tak aby procesy były bardziej efektywne i konkurencyjne.

Ale ważne aby odnotować zmianę w celach, które cele są obecnie najważniejsze: Dzisiejsze systemy SCM są całkowicie ukierunkowane na klienta. Celem zarządzania łańcuchem dostaw zawsze było zwiększanie efektywności i obniżanie kosztów. Cele te nie uległy zmianie, ale obecnie główną rolę w ustalaniu priorytetów takiego zarządzania odgrywa klient. Mówi się, że „doświadczenia klientów żyją i umierają w łańcuchu dostaw”. Lojalność klientów zależy od tego, czy przedsiębiorstwo jest w stanie szybko i dokładnie spełnić ich oczekiwania. Dostawy surowców, produkcja, logistyka, sprzedaż i zarządzanie zamówieniami muszą być ze sobą skoordynowane, aby klient otrzymał to, co chce, w rozsądnym terminie. W tym celu przedsiębiorstwo musi analizować swoje łańcuchy dostaw z perspektywy klientów. Nie chodzi tu bowiem tylko o terminową dostawę, ale również o wykonanie we właściwym czasie wszystkich niezbędnych czynności, zarówno przed taką dostawą, jak i w jej trakcie oraz po jej zrealizowaniu. [10]

2.1.3. Znaczenie zarządzania Łańcuchem Dostaw

Łańcuchy dostaw zawsze były napędzane przez wiele sił globalnych i politycznych, a nawet przez warunki pogodowe i wydarzenia naturalne. Jest jednak jedna rzecz, która jest pewna w zarządzaniu łańcuchem dostaw i to jest zmiana. Nowoczesna logistyka stawia menedżerów przed wieloma wyzwaniami skutecznej organizacji łańcuchów dostaw, optymalizacji kosztów i zarządzania stanami magazynowymi. Wszystko to ułatwiają rozwijające się technologie Łańcuchów Dostaw 4.0. To szerokie pojęcie obejmuje między innymi zastosowanie IoT w logistyce, użycie zaawansowanej robotyki w magazynach oraz zaawansowaną analizę Big Data, a nawet sztuczną inteligencję. Dzięki czujnikom, sieciom, nowoczesnemu oprogramowaniu i automatyzacji możliwe jest nie tylko zwiększenie skuteczności procesów logistycznych, ale także ograniczenie kosztów oraz wyższy poziom

zadowolenia klienta z dostaw. Znaczenie systemów SCM: Istota zarządzania łańcuchem dostaw Wystarczy się rozejrzeć. Wszystko, co znajduje się w domu lub zakładzie pracy, trafiło tu dzięki łańcuchom dostaw. Działania te wymagają zaangażowania pracowników z milionów miejsc pracy na całym świecie. Przez łańcuch dostaw przewija się wszystko — od tanich dóbr konsumpcyjnych aż po przyrządy chirurgiczne i kluczowe zasoby. Wciąż jednak, mimo iż zarządzanie łańcuchami dostaw leży u podstaw gospodarki światowej, wiele organizacji korzysta w tym celu z procesów i maszyn stosowanych od 50 lat.[7]

Wiele firm po prostu przestałoby działać bez zarządzania łańcuchem dostaw, więc to dość duże korzyści SCM. Niektóre z zalet zoptymalizowanego zarządzania łańcuchem dostaw obejmują:

- Większa produktywność: systemy do zarządzania zasobami przedsiębiorstwa i przeglądy zapobiegawcze zwiększają wydajność maszyn i systemów. Może to pomóc w wyeliminowaniu wąskich gardeł, usprawnieniu procesów i zwiększeniu produktywności. Zautomatyzowane procesy i responsywne analizy danych przekładają się na szybszy transport i krótszy czas dostawy.

- Niższe koszty łańcucha dostaw: przy użyciu analiz predykcyjnych można wyeliminować konieczność szacowania w oparciu o przypuszczenia. Pozwoli to ograniczyć zarówno marnotrawienie zapasów, jak i ryzyko ich deficytu. Internet rzeczy zapewnia większą responsywność istniejących zasobów oraz maksymalnie wydajny i praktyczny przepływ pracy w każdej sytuacji. Bardziej precyzyjne prognozy umożliwiają ponadto redukcję liczby niezaladowanych w pełni samochodów dostawczych i nieskoordynowanych tras dostaw, a także bardziej efektywne zarządzanie flotą.

- Większa elastyczność i odporność łańcucha dostaw: Trendy i zmiany na rynku mogą nastąpić nagle. Odporne systemy SCM są elastyczne, aby dostosować się do każdej sytuacji. Dane w czasie rzeczywistym i inteligentne analizy mogą pomóc menedżerom łańcucha dostaw przenieść maszyny i personel do lepszych przepływów pracy. Opinie klientów można od razu usłyszeć i podjąć odpowiednie działania. Wirtualne zapasy i inteligentne procesy magazynowe zapewniają dostosowanie podaży i popytu.

- Wyższa jakość produktów: zapewnienie zespołom ds. badań i rozwoju dostępu do informacji zwrotnych od klientów sprawia, że podczas projektowania i opracowywania produktów brane są pod uwagę ich potrzeby. Zarówno zespół ds. badań i rozwoju, jak i zespół ds. produkcji mogą wykorzystać informacje dostępne dzięki uczeniu maszynowemu i analizom, aby reagować na trendy i oczekiwania klientów oraz doskonalić projekty produktów.

- Lepsza obsługa klienta: najlepsze praktyki SCM są zorientowane na klienta i zaprojektowane tak, aby były responsywne i adaptacyjne. Dzięki konkurencji tylko jednym kliknięciem, nowoczesny SCM pozwala firmom wdrażać opinie klientów i trendy, umożliwiając zarówno mikrorealizację, jak i personalizację na dużą skalę.

- Większa przejrzystość i zrównoważony rozwój: SCM umożliwia pełną przejrzystość,

od etapu projektowania i produkcji po logistykę, dostawę i zwroty. Dzięki możliwości wglądu we wszystkie dane wejściowe i wyjściowe w całym łańcuchu organizacje mogą znacznie poprawić swój wpływ na środowisko, często współpracując w tym celu bezpośrednio z dostawcami i innymi dostawcami. [7]

2.1.4. Składniki Efektywnego Zarządzania Łańcuchem Dostaw

Elementy łańcucha dostaw Istnieje pięć elementów tradycyjnych systemów zarządzania łańcuchem dostaw:

- Planowanie. Planowanie i zarządzanie wszystkimi zasobami wymaganymi do zaspokojenia zapotrzebowania klientów na produkt lub usługę firmy. Po ustanowieniu łańcucha dostaw należy określić mierniki, które pozwolą zmierzyć czy łańcuch dostaw jest wydajny, efektywny, zapewnia wartość klientom i spełnia cele firmy.

- Zaopatrzenie. Wybór dostawców kwalifikowanych, którzy zapewnią towary i usługi potrzebne do stworzenia produktu. Należy również ustalić procesy monitorowania i zarządzania relacjami z dostawcami. Kluczowe działania w tych procesach obejmują zamawianie, przyjmowanie, zarządzanie stanami magazynowymi oraz realizację płatności dla dostawców.

- Produkcja. Czynności wymagane do przyjęcia surowców, wytworzenia produktu, badania jakości, opakowania do wysyłki i przygotowania harmonogramu dostaw.

- Dostawa i logistyka. Koordynacja zamówień klientów, magazynowanie, planowanie dostaw, wysyłka towarów, wystawianie faktur i odbieranie płatności.

- Zwroty. Stworzenie procesu odbioru wadliwych, nadmiarowych lub niechcianych produktów[8]

Aleksander Wielki powiedział kiedyś słynnie: „Moi logistycy nie mają poczucia humoru... ponieważ wiedzą, że jeśli moja kampania się nie powiedzie, są pierwszymi, których zabiję”. I choć ten przykład jest trochę skrajny, niemniej jednak ilustruje on, jak ważne dla cywilizacji ludzkiej były zawsze łańcuchy dostaw. Efektywne i odporne narzędzia i praktyki w zakresie zarządzania łańcuchem dostaw są niezbędnym elementem przetrwania i sukcesu firmy. Niektóre z podstawowych procesów SCM obejmują:

- Planowanie łańcucha dostaw to proces przewidywania popytu na produkty i koordynowania powiązań w łańcuchu dostaw w celu jego realizacji. Oprócz prognozowania popytu i planowania obejmuje on planowanie dostaw, planowanie potrzeb materiałowych (MRP), planowanie produkcji, planowanie sprzedaży i produkcji (SOP) i inne.

- Zarządzanie cyklem życia produktu to proces zarządzania produktem przez cały jego cykl życia – od pomysłu, inżynierii i projektowania po produkcję, serwis i utylizację (lub recykling).

- Nabycie to proces nabywania materiałów, towarów i usług w celu zaspokojenia potrzeb biznesowych i zapewnienia jakości, uczciwej ceny i wartości tych towarów. Głównym

wyzwaniem dla zaopatrzenia i określania źródła dostaw jest przewidywanie dokładnych wolumenów zamówień, ponieważ zarówno niedobory, jak i nadwyżki mogą być szkodliwe dla firmy.

- Zarządzanie logistyką to transport i składowanie towarów od początku łańcucha dostaw, z surowcami i produkcją, po dostawę gotowych produktów do sklepów lub klientów – a nawet do obsługi produktów, zwrotów i recyklingu. Funkcje biznesowe obejmują zarządzanie transportem przychodzącym i wychodzącym, zarządzanie flotą, gospodarkę magazynową, kontrolę zapasów i obsługę klienta.

- Zarządzanie realizacją produkcji (MES) monitoruje, śledzi, dokumentuje i kontroluje proces wytwarzania towarów. Utrzymuje produkcję i procesy tak szczupłe, jak to tylko możliwe – zachowując (i poprawiając) jakość, zrównoważony rozwój i zadowolenie klientów.

- Zarządzanie aktywami przedsiębiorstwa to proces zarządzania aktywami fizycznymi i ich utrzymania w całym łańcuchu dostaw, od robotyki fabrycznej po floty dostaw. Czujniki IoT, łączność maszyna-maszyna (M2M) – poprawiają wydajność, czas pracy, bezpieczeństwo oraz prewencyjną i predykcyjną konserwację. Niektóre połączone zasoby mogą nawet przewidywać naprawy lub awarie i samodzielnie przeprowadzać konserwację – bezpośrednio do określania źródła dostaw i zamawiania części potrzebnych do wydłużenia cyklu życia.[7]

2.2. Maszynowe Uczenie

W tym rozdziale omawiamy podstawowe koncepcje i techniki związane z maszynowym uczeniem. Rozpoczynamy od wprowadzenia do sztucznej inteligencji, definiując, czym jest i jakie są jej główne komponenty. Następnie skupiamy się na maszynowym uczeniu, jego znaczeniu i zastosowaniach.

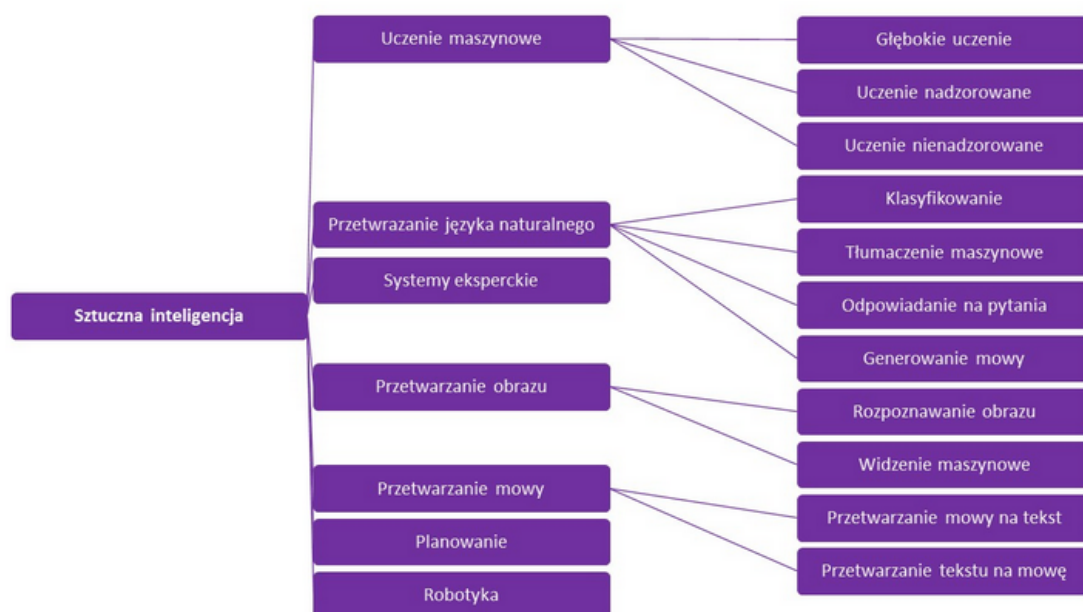
2.2.1. Sztuczna Inteligencja (SI)

Sztuczna inteligencja to dziedzina informatyki, która zajmuje się tworzeniem systemów komputerowych zdolnych do wykonywania zadań, które normalnie wymagają ludzkiego myślenia. SI składa się z kilku głównych komponentów, obejmujących:

Definicje sztucznej inteligencji Sztuczna inteligencja jest tematem obszernym i szeroko omawianym zarówno w sferze naukowej, publicystycznej, jak i politycznej. Są to działania oparte o modelowanie wiedzy, danych i rozwijanie systemów algorytmów oraz mocy obliczeniowych, co w obecnym stanie techniki pozwala na uzyskanie względnie zautomatyzowanego systemu pozyskiwania, przetwarzania i analizy danych, który daje możliwość samoistnego (autonomicznego) ulepszania systemu lub przewidywania zachowań i działań na podstawie analizy zebranych danych i korelacji między nimi, z możliwością wpływu na środowisko zewnętrzne oraz pozostające z nim w interakcji za pomocą sensorów i siłowników. Interakcje te mogą zachodzić mechanicznie lub z udziałem czło-

wieka w cyklu życia sztucznej inteligencji począwszy od etapu kreacji, rozwoju, wdrożenia, stosowania, aż po etap decyzji o wyłączeniu z pracy i utylizacji.[11]

Ponizej podział i schemat zastosowań sztucznej inteligencji wg. raportu "Game-changing technologies: Transforming production and employment in Europe"



Rysunek 2.1. Obszary zastosowań sztucznej inteligencji[11]

Definicja wg.raportu: "A DEFINITION OF AI:MAIN CAPABILITIES AND DISCIPLINES" Sztuczna inteligencja (AI) odnosi się do systemów, które wykazują inteligentne zachowanie poprzez analizę swojego otoczenia i podejmowanie działań – z pewną dozą autonomii – dla osiągnięcia określonych celów. Systemy oparte na AI mogą mieć charakter wyłącznie programowy, działać w świecie wirtualnym (np. asystenci głosowi, analiza obrazu oprogramowanie, wyszukiwarki, systemy rozpoznawania mowy i twarzy) lub sztuczna inteligencja mogą być wbudowane w urządzenia sprzętowe (np. zaawansowane roboty, samochody autonomiczne, drony czy aplikacje Internetu Rzeczy[12]

- **Uczenie maszynowe (Machine Learning):** Jest to jedna z najważniejszych gałęzi sztucznej inteligencji, która pozwala komputerom na naukę z danych i podejmowanie decyzji na podstawie tych informacji. W ramach uczenia maszynowego wykorzystuje się różne techniki, w tym algorytmy uczenia nadzorowanego i nienadzorowanego.
- **Przetwarzanie języka naturalnego (Natural Language Processing - NLP):** Ta dziedzina zajmuje się rozumieniem, przetwarzaniem i generowaniem języka naturalnego przez komputery. NLP jest używane do analizy tekstu, tłumaczenia maszynowego i wielu innych zastosowań.

- **Wizja komputerowa:** Obejmuje technologie pozwalające komputerom analizować, rozumieć i interpretować obrazy i wideo. Jest używana w rozpoznawaniu obiektów, rozpoznawaniu twarzy, samochodach autonomicznych i innych dziedzinach.
- **Robotyka:** SI jest również istotna w dziedzinie robotyki, gdzie komputery sterują działaniami fizycznymi robotów.

2.2.2. Maszynowe Uczenie

Maszynowe uczenie (ML) to podzbiór sztucznej inteligencji, który koncentruje się na rozwijaniu algorytmów i technik pozwalających komputerom na naukę z danych i podejmowanie decyzji na podstawie tych informacji. Głównym celem maszynowego uczenia jest rozwijanie modeli, które mogą generalizować zbiory danych i wykonywać zadania bez konieczności programowania ich wprost.

2.2.3. Zastosowania Maszynowego Uczenia

Maszynowe uczenie ma szerokie spektrum zastosowań w różnych dziedzinach. W kontekście zarządzania łańcuchem dostaw (SCM), ma to szczególne znaczenie. Poniżej przedstawiamy niektóre z głównych zastosowań ML w SCM:

- **Prognozowanie popytu:** Uczenie maszynowe może być wykorzystywane do prognozowania przyszłego popytu na produkty lub usługi, co pomaga w planowaniu produkcji i zarządzaniu zapasami.
- **Optymalizacja zapasów:** Algorytmy ML mogą pomóc w zoptymalizowaniu poziomu zapasów, minimalizując koszty i zapewniając, że dostępność produktów jest na odpowiednim poziomie.
- **Wybór dostawcy:** ML może pomóc w analizie dostawców pod kątem efektywności, jakości i kosztów, co ułatwia wybór najlepszego dostawcy.
- **Planowanie tras i logistyka:** Algorytmy ML mogą pomóc w optymalizacji tras dostaw, minimalizując czas i koszty transportu.

2.2.4. Rodzaje Maszynowego Uczenia

Maszynowe uczenie może być podzielone na kilka głównych rodzajów, zależnie od sposobu przetwarzania danych i celu uczenia. Oto niektóre z najważniejszych rodzajów maszynowego uczenia:

- **Uczenie nadzorowane (Supervised Learning):** W tym rodzaju uczenia algorytm jest trenowany na podstawie zestawu danych, który zawiera zarówno wejścia, jak i odpowiadające im oczekiwane wyjścia. Celem jest zbudowanie modelu, który może dokładnie przewidywać wyjście na podstawie nowych danych wejściowych. Przykłady obejmują algorytmy regresji liniowej i klasyfikacji.
- **Uczenie nienadzorowane (Unsupervised Learning):** W przypadku uczenia nienadzorowanego algorytm jest trenowany na danych wejściowych bez oczekiwanych

wyjąć. Celem jest odkrywanie ukrytych wzorców, grupowanie danych lub redukcja wymiarowości. Przykładem jest klastrowanie danych.

- **Uczenie ze wzmocnieniem (Reinforcement Learning):** Ten rodzaj uczenia polega na trenowaniu agenta, który podejmuje decyzje w środowisku w celu maksymalizacji nagrody. Agent eksploruje różne działania i dostaje informacje zwrotną w postaci nagród lub kar za swoje decyzje. Jest używany w dziedzinach takich jak gry komputerowe i robotyka.
- **Uczenie pół-nadzorowane (Semi-supervised Learning):** Uczenie pół-nadzorowane łączy elementy uczenia nadzorowanego i nienadzorowanego. Model jest trenowany na części danych z etykietami i części danych bez etykiet. Jest stosowane, gdy dostępne są tylko częściowo oznaczone dane.
- **Uczenie głębokie (Deep Learning):** To rodzaj maszynowego uczenia opartego na sieciach neuronowych o wielu warstwach. Jest stosowane do złożonych problemów przetwarzania obrazów, dźwięku i tekstu. Deep Learning osiągnął ogromny sukces w dziedzinach takich jak rozpoznawanie obrazów i przetwarzanie języka naturalnego.

Wybór odpowiedniego rodzaju maszynowego uczenia zależy od konkretnego problemu i dostępnych danych. W kontekście zarządzania łańcuchem dostaw, różne rodzaje uczenia maszynowego mogą być stosowane w zależności od konkretnych zadań, takich jak prognozowanie popytu czy optymalizacja tras dostaw.

2.3. Uczenie maszynowe w zarządzaniu łańcuchem dostaw

W tym rozdziale dokładniej analizujemy, w jaki sposób maszynowe uczenie może być wykorzystane w zarządzaniu łańcuchem dostaw. Przedstawiamy przykłady konkretnych zastosowań uczenia maszynowego w SCM, takie jak prognozowanie popytu, optymalizacja zapasów czy wybór dostawcy. Omawiamy również korzyści wynikające z wykorzystania tych technik oraz wyzwania, jakie mogą się pojawić.

2.3.1. Prognozowanie popytu

Jednym z kluczowych elementów zarządzania łańcuchem dostaw jest zdolność do dokładnego prognozowania popytu na produkty. Maszynowe uczenie oferuje zaawansowane metody analizy danych historycznych oraz czynników wpływających na popyt, co umożliwia dokładniejsze prognozowanie przyszłego zapotrzebowania. Metody te obejmują:

- **Modele szeregów czasowych:** Wykorzystują dane historyczne do identyfikowania wzorców w sezonowości i trendach, co pozwala na prognozowanie przyszłego popytu na podstawie danych z przeszłości.
- **Algorytmy uczenia maszynowego, takie jak sieci neuronowe i drzewa decyzyjne:** Mogą analizować bardziej skomplikowane wzorce w danych, uwzględniając różnorodne zmienne i interakcje między nimi.

- Analiza sentymentu i danych społecznościowych: Wykorzystuje opinię klientów i informacje pochodzące z mediów społecznościowych do oceny wpływu opinii publicznej na popyt na produkty.

Precyzyjne prognozy popytu pozwalają firmom lepiej zarządzać swoimi zapasami, minimalizować ryzyko nadmiernego lub niewystarczającego stanu magazynowego oraz dostosować produkcję do rzeczywistych potrzeb rynku.

Modele szeregów czasowych Szereg czasowy to ciąg uporządkowanych obserwacji, których dokonuje się w określonych (zwykle stałych) jednostkach czasu. Szeregiem czasowym będzie na przykład średnia cena energii elektrycznej w poszczególnych miesiącach lub ilość zgłaszanych przypadków przemocy domowej w poszczególnych latach. Szereg czasowy można przedstawić w formie wykresu lub tabeli. Wyróżniamy dwa podstawowe składniki szeregu czasowego – trend i sezonowość. Pojęcie trendu odnosi się do długotrwałej i systematycznej zmiany wielkości danego zjawiska, natomiast sezonowości do rytmicznych, powtarzających się zmian wielkości danego zjawiska, charakteryzujące się różną długością całego cyklu. Celem analizy szeregów czasowych jest identyfikowanie i analiza jego struktury, jak również prognozowanie – przewidywanie wartości w kolejnych odstępach czasu, np. sprzedaży. Istnieje wiele metod analizy szeregów czasowych, z czego wiele z nich jest złożonych i obejmują one m.in. analizę trendu (techniki wygładzania) i sezonowości (badanie struktury autokorelacji), modelowanie i prognozowanie szeregów czasowych (m.in. metody ARIMA, wyrównywanie wykładnicze, analiza harmoniczna, analiza widma).[13]

Szereg czasowy jest bardzo często wykreślany za pomocą wykresu przebiegu (który jest wykresem liniowym czasowym). Szeregi czasowe są wykorzystywane w statystyce, przetwarzaniu sygnałów, rozpoznawaniu wzorców, ekonometrii, finansach matematycznych, prognozowaniu pogody, przewidywaniu trzęsień ziemi, inżynierii sterowania, astronomii, inżynierii komunikacji i głównie w każdej dziedzinie nauk stosowanych i inżynierii, która obejmuje pomiary czasowe. Analiza szeregów czasowych obejmuje metody analizy danych szeregów czasowych w celu wyodrębnienia znaczących statystyk i innych cech charakterystycznych danych. Prognozowanie szeregów czasowych polega na wykorzystaniu modelu do przewidywania przyszłych wartości na podstawie wcześniej zaobserwowanych wartości. Chociaż analizę regresji często wykorzystuje się w celu sprawdzenia zależności między jednym lub większą liczbą różnych szeregów czasowych. W kontekście statystyki, ekonometrii, finansów ilościowych, sejsmologii, meteorologii i geofizyki głównym celem analizy szeregów czasowych jest prognozowanie. W kontekście przetwarzania sygnałów, inżynierii sterowania i inżynierii komunikacji służy do wykrywania sygnału. Inne zastosowania obejmują eksplorację danych, rozpoznawanie wzorców i uczenie maszynowe, gdzie analizę szeregów czasowych można wykorzystać do grupowania, (uwaga od artykułu źródła!!)klasyfikacji, zapytań według treści, wykrywania anomalii i prognozowania.[14]

Poniżej znajdują się elementy szeregu czasowego:

- Sezonowość: Ten element jest powiązany z kalendarzem, stałe lub okresowe wachania szeregu czasowego zmiennego w czasie, istnieje a powtarzający się wzór.
- Trend: Ten komponent odnosi się do długoterminowego trendu wartości obserwacji które z czasem mogą się zwiększać lub zmniejszać.
- Cykliczny: Ten składnik odnosi się również do okresowych wachnię czasu serii, ale w odróżnieniu od składnika sezonowości nie jest on stały
- Losowy: Po wyodrębnieniu wszystkich trzech pozostałych składników z czasu w serii pozostaje element losowy, zwykle ma on strukturę, którą należy zidentyfikować za pomocą modelu w celu dokładnego prognozowania przyszłej wartości szeregu czasowego. [15]

Mozna wyróżnić wiele modeli szeregów czasowych stosowane do prognozowania :

Rodzaje modeli szeregów czasowych	
Typ szeregu czasowego	Modele szeregów czasowych
Szeregi stacjonarne	Naiwne
	Średniej ruchomej
	Prosty model wygładzania wykładniczego
	Modele ARMA i ARIMA
Szeregi wykazujące tendencje rozwojowa. (trend)	Wygładzanie wykładnicze
	Model liniowy Holta
	modele analityczne
	modele adaptacyjne
Szeregi wykazujące sezonowość	model wintorsa
	metoda wskaźników analiza harmoniczna

Rysunek 2.2. s1a Modele szeregów czasowych[16]

Modele szeregów stacjonarnych. Stacjonarność oznacza, że statystyczne właściwości szeregu czasowego, takie jak średnia i wariancja, są stałe w czasie lub nie zmieniają się w sposób istotny. Modele stacjonarne zakładają, że szereg czasowy ma ustalone właściwości, co ułatwia analizę i prognozowanie. Istnieje kilka popularnych modeli szeregów stacjonarnych, w tym:

Model ARMA (AutoRegressive Moving Average): Model ten łączy dwie podstawowe komponenty - autoregresję (AR), która uwzględnia zależności między obserwacjami w czasie, oraz średnią ruchomą (MA), która uwzględnia wpływ losowych zakłóceń.

Autoregresja to jest model stacjonarny który do predykcji przyszłych wartości używa: stalej + przeszłych danych (np jak w modelu AR może być 2 rzędu wówczas 2 ostatnich pomiarów) + współczynnika autoregresji na podstawie danych historycznych, im wyższy tym ostatnie dane mają większy wpływ na wynik + współczynnik błędu [17][18]

$$Y_t = C + \Theta \cdot Y_{t-1} + E_t$$

Model autoregresji[19]

To znaczy, zgodnie z modelem AR (1), zmienna y w czasie t jest równa stałej (c), plus zmienna w $(t-1)$ pomnożona przez współczynnik plus błąd. Należy zauważyć, że stała „ c ” może być liczbą dodatnią, ujemną lub zerową. Jeśli chodzi o wartość teta, czyli współczynnik pomnożony przez $y(t-1)$, może przyjmować różne wartości.

MODEL ARIMA, czyli AutoRegressive Integrated Moving Average, to bardziej zaawansowany model szeregów czasowych, który łączy w sobie trzy główne składniki: model

autoregresji (AR), model średniej ruchomej (MA) i różnicowanie (I, od ang. Integrated). Jest modelem powstałym na bazie ARMA. W przypadku braku różnicowania model przekształca się w ARMA.[20][21]

Metoda naiwna – metoda prognostyczna dotycząca analizy szeregów czasowych bez tendencji. Metoda ta stosowana jest przy stałym poziomie zjawiska i niewielkich wahaniami przypadkowych. Zalety:

- prosta i łatwa do zrozumienia
- szybka i tania

Wady:

- niska jakość prognoz

Jest to prognoza typu: "jutro będzie tak jak dziś"[22][15]

Mogą być stosowane w przypadku stwierdzenia niewielkich wahań w szeregu zmiennej prognozowanej i wykorzystane w prognozowaniu krótkookresowym, obejmującym jeden okres naprzód. Metody naiwne są szybkie i tanie w zastosowaniu, jednak jakość prognoz wyznaczonych z ich użyciem jest zwykle niska. Przykłady: sprzedaż przedsiębiorstwa w następnym kwartale będzie na dotychczasowym poziomie, zysk wzrośnie w tym samym stopniu co ubiegłym miesiącu.[16]

Metoda naiwna, algorytm prognozowania

$$Y^*_t = Y_{t-1}$$

gdzie Y^*_t prognoza zmiennej Y dla momentu t , Y_{t-1} to obserwacja rzeczywistej wartości zmiennej Y dla chwili $t-1$, Metoda naiwna[16]

Metoda średniej ruchomej: Polega na zastępowaniu danych empirycznych dla kolejnych okresów średnimi poziomami z okresu badanego i kilku okresów sąsiednich. Metoda średniej ruchomej może być stosowana zarówno do wygładzania szeregu czasowego, jak i do prognozowania.

$$Y_t = \frac{1}{k} \sum_{i=t-k}^{t-1} Y_i$$

gdzie Y_t to prognoza zmiennej Y dla momentu t , Y_i obserwacja rzeczywistej wartości zmiennej Y dla chwili i , k to liczba ruchomych składników szeregu czasowego

średnia ruchoma algorytm[16]

Zalety: prosty algorytm, szybkie i tanie prognozowanie

Wady: konieczność doboru stałej K (min. ilości błędów) dla dużego K potrzeba trzymania dużej ilości danych Dla usprawnienia działania można dodać czynniki wag do algorytmu to znaczy że okresy mają różną wartość (trochę komplikuje algorytm). Jest to prognoza krótkookresowa, typowa stacjonarna gdzie zakładamy że prognozowany okres będzie podobny do ostatnich k - okresów[16]

Model wykładzania wykładniczego. Wygładzanie wykładnicze (ang. exponential smoothing) – metoda obróbki szeregu czasowego zmniejszająca jego wariancję za pomocą ważonej średniej ruchomej z przeszłych wartości, o wagach malejących wykładniczo wraz z odległością w czasie. Stosowana do prostego usuwania szumu (Stosunek sygnału do

szumu (SNR, ang. signal-to-noise ratio) – miara porównująca poziom sygnału użytecznego (informacja) do poziomu szumu tła (niepożądany sygnał). Jest definiowana jako stosunek mocy sygnału użytecznego do mocy szumu tła i jest często wyrażona w decybelach (dB). Jest przydatna w prognozowaniu szeregów czasowych o niewielkim stosunku sygnału do szumu, szczególnie niemających wyraźnego trendu i wahań sezonowych. [23]

Jest rozszerzeniem idei średniej ruchomej i polega na:wygładzeniu oryginalnego szeregu tak, jak robi to średnia ruchoma ,użycia otrzymanego szeregu do uzyskania przyszłych wartości zmiennej, Istotne jest zwiększenie wpływu ostatnich wartości szeregu na prognozę, w stosunku do wpływu bardziej odległych obserwacji ,Największa waga jest nadana bieżącej obserwacji, mniejsza waga poprzedniej obserwacji i tak dalej. Wagi zmniejszają się geometrycznie w miarę cofania się w czasie. Jest to prosty i szybki algorytm typowy do danych stacjonarnych i nadaje się na krótkookresowe prognozy[16]

$$\begin{aligned} Y'_t &= \alpha \cdot Y_{t-1} + (1 - \alpha) \cdot Y'_{t-1} \\ &= Y'_{t-1} + \alpha \cdot (Y_{t-1} - Y'_{t-1}) \end{aligned}$$

gdzie Y'_t to prognoza zmiennej Y dla momentu t , Y_{t-1} to obserwacja rzeczywistej wartości zmiennej Y dla chwili $t-1$, α to parametr usrdniania (wygładzania) z przedziału $[0,1]$

Wygładzanie wykładnicze algorytm[16]

Kolejna grupa metod to włączające w prognoze trend należa do nich: metoda liniowa Holta, modele analityczne i adaptacyjne

Model Holta stosujemy wygładzając szereg czasowy, w którym wyróżnić można zarówno trend (tendencję rozwojową) jak i wachania przypadkowe. występują dwa parametry: alfa i beta.[24] Model Holta pozwala na wygładzanie szeregu czasowego, w którym

występuje tendencja rozwojowa oraz wachania przypadkowe. Wartości prognozowanego szeregu zostały oznaczone symbolami x_0, x_1, \dots, x_{n-1} . Model ten ma dwa parametry alfa oraz beta i następującą postać:

$$\begin{aligned} F_t &= \alpha \cdot X_t + (1 - \alpha) \cdot (F_{t-1} + S_{t-1}) \\ S_t &= \beta \cdot (F_t - F_{t-1}) + (1 - \beta) \cdot S_{t-1} \end{aligned}$$

gdzie $t = 2, \dots, n-1$,

F_t - wygładzona wartość szeregu czasowego,

S_t - wygładzona wartość przyrostu trendu na moment t ,

α, β - parametry modelu

Model Holta obliczanie składników szeregu czasowego i trendu[25]

Parametry modelu Holta alfa oraz beta są dobierane tak, aby zminimalizować błędy prognoz[25]

Podwójne wygładzanie wykładnicze –Metoda HOLT Przystosowane do szeregów czasowych ze składnikami sezonowymi i trendami. Ogólna idea polega na tym, że prognozy są obliczane nie tylko na podstawie kolejnych poprzednich obserwacji (jak w prostym wyrównywaniu wykładniczym), ale można także dodać niezależny trend i składnik sezonowy. Wiele empirycznych szeregów czasowych zawiera wachania sezonowe. (np. roczna sprzedaż komputerów osiąga prawdopodobnie szczyt w listopadzie i grudniu i być może latem) Wzorzec ten będzie się prawdopodobnie powtarzał co roku, chociaż względna wielkość wzrostu sprzedaży w grudniu może się powoli zmieniać z roku na rok. Metoda nadaje się do krótkookresowej prognozy. Zalety elastyczność zaś wady trudność doboru parametrów[16]

algorytm prognozowania:

równanie 1 (usrednienie szeregu czasowego)

$$F_{t-1} = \alpha \cdot Y_{t-1} + (1 - \alpha) \cdot (F_{t-1} + S_{t-1})$$

równanie 2 (usrednianie trendu)

$$S_t = \beta \cdot (F_t - F_{t-1}) + (1 - \beta) \cdot S_{t-1}$$

równanie 3 (wyznaczenie prognozy):

$$Y'_t = F_n + (T - n) \cdot S_n$$

gdzie F_{t-1} to usredniona prognoza zmiennej Y dla chwili $t-1$, S_{t-1} to usredniony trend zmiany wartości zmiennej Y dla chwili $t-1$, α, β to parametry usredniania z przedziału $[0,1]$, Y'_t to prognoza zmiennej Y dla chwili t , n to liczba elementów szeregu czasowego

Model Holta cały algorytm[16]

Kolejny przykład: Potrójne wygładzanie wykładnicze Metoda Wintersa

Model Holta-Wintersa działa na zasadzie „wygładzania” danych szeregów czasowych za pomocą trzech równań wygładzania: dla poziomu, trendu i sezonowości. W efekcie model jest w stanie uwzględnić zmiany zarówno w trendzie, jak i w sezonowości, dostarczając dokładniejsze prognozy.[26] Model Wintersa może być stosowany w przypadku szere-

gów czasowych zawierających tendencję rozwojową, wahania sezonowe oraz wahania przypadkowe. [27]

$$L_{t-1} = \alpha \cdot \frac{Y_{t-1}}{C_{t-1-r}} + (1 - \alpha) \cdot (L_{t-2} + B_{t-2})$$

$$B_{t-1} = \beta \cdot (F_{t-1} - F_{t-2}) + (1 - \beta) \cdot B_{t-2}$$

$$S_{t-1} = \gamma \cdot \frac{Y_{t-1}}{F_{t-1}} + (1 - \gamma) \cdot S_{t-1} - r$$

gdzie:

L_{t-1} odpowiednik wygładzonej wartości otrzymanej z prostego modelu wygładzania wykładniczego (ocena wartości średniej)

B_{t-1} ocena przyrostu trendu na moment lub okres $t-1$

S_{t-1} ocena wskaźnika sezonowości na moment lub okres $t-1$

r długość cyklu sezonowego

a, b i c parametry modelu przyjmujące wartości z przedziału $[0,1]$

Model Wintersa obliczane składniki [27]

pełny algorytm:

Potrójne wygładzanie wykładnicze

Metoda Wintersa

$$F_t = \alpha \cdot (Y_t - S_{t-r}) + (1 - \alpha) \cdot (F_{t-1} + T_{t-1})$$

$$T_t = \beta \cdot (F_t - F_{t-1}) + (1 - \beta) \cdot T_{t-1}$$

$$S_t = \gamma \cdot (Y_t - F_t) + (1 - \gamma) \cdot S_{t-r}$$

$$Y'_t = F_n + (\tau - n) \cdot T_n + S_{t-1}$$

$$\tau > n$$

F_t wygładzona wartość zmiennej prognozowanej

$T_t = S_t$ wygładzona wartość przyrostu trendu

$S_t = C_t$ ocena wskaźnika sezonowości na moment t

α stała wygładzania poziomu zmiennej F

β stała wygładzania współczynnika trendu T

γ stała wygładzania efektu sezonowego S

r długość cykli sezonowości

n numer ostatniej obserwacji

Model Wintersa pełny algorytm [16]

Algorytmy uczenia maszynowego w maszynowym uczeniu można wyodrębnić:

Uczenie nadzorowane (ang. supervised learning) – sposób uczenia, w którym zbiór danych treningowych, na których uczy się algorytm, zawiera dołączone rozwiązanie problemu, tzw. etykiety albo klasy.

Uczenie nienadzorowane(ang. unsupervised learning) – sposób uczenia modelu, w którym dane uczące są nieoznakowane tzn. nie posiadają etykiet. Mówiąc potocznie, posiadamy surowe dane które wrzucamy do modelu i zostawiamy algorytmowi całą pracę związaną ze znalezieniem powiązań między danymi (tzw. uczenie bez nauczyciela).

Uczenie przez wzmacnianie(ang. RL - reinforcement learning) – jeden z trzech głównych nurtów uczenia maszynowego, którego zadaniem jest interakcja ze środowiskiem na podstawie zebranych informacji. W przeciwieństwie do wymienionych wcześniej rodzajów, w uczeniu przez wzmacnianie nie przygotowuje się zestawu danych uczących, tylko środowisko, z którego model będzie zbierał dane automatycznie.

Uczenie półnadzorowane(ang. semisupervised learning) – algorytmy działające na częściowo oznakowanych danych[28]

Algorytmy prognozowania ML często wykorzystują techniki obejmujące bardziej złożone cechy i metody predykcyjne, ale cel metod prognozowania ML jest taki sam, jak w przypadku metod tradycyjnych – poprawa dokładności prognoz przy jednoczesnej minimalizacji funkcji straty. Funkcję straty przyjmuje się zwykle jako sumę kwadratów spowodowaną błędami w przewidywaniu/prognozowaniu. Najważniejsza różnica pomiędzy obiema metodami polega na sposobie minimalizacji. Chociaż większość tradycyjnych metod wykorzystuje wyjaśnialne procesy liniowe, większość metod ML wykorzystuje techniki nieliniowe w celu minimalizacji funkcji strat.

Oto kilka przykładów modeli prognozowania uczenia maszynowego stosowanych w aplikacjach biznesowych:

- Artificial neural network
- Long short-term-memory-based neural network
- Random forest
- Generalized regression neural networks
- K-nearest neighbors regression
- Classification and regression trees (CART)
- Support vector regression
- Gaussian processes [29]

w literaturze poświęconej stosowaniu maszynowego uczenia w prognozowaniu popytu spotkałem się ze najczęściej stosowane modele do prognozowania popytu to:

Neural Networks (NN), zwane również sztucznymi sieciami neuronowymi, są inspiracją biologicznych neuronów. Składają się z warstw neuronów (lub węzłów), które przetwarzają dane i przekazują informacje między sobą za pomocą wag.

Recurrent Neural Networks (RNN) to rodzaj sieci neuronowych, które posiadają pętle

rekurencyjne, które pozwalają im na uwzględnianie informacji z poprzednich kroków czasowych w analizie sekwencji danych..

Support Vector Machines (SVM) to technika uczenia maszynowego stosowana zarówno do zadań klasyfikacji, jak i regresji. Tutaj jest wybierana hiperpłaszczyzna decyzyjna które najlepiej oddziela klasy i pozwala prognozować (na podstawie danych treningowych i cech), optymalizuje margines (miedzy najbliższymi punktami danych a hiperpłaszczyzną wektory nosne), używa funkcje jądrowe które umowzliwiają przekształcanie danych do wyższych wymiarów/przestrzeni gdzie zostają liniowo odseparowane (mimo że zależności nie są liniowe). Najbardziej popularne to : RBF (Radial Basis Function) i funkcja wielomianowa. Dobierane są parametry np błęd które powodują zwiększenia lub zmniejszenie czułości na dokładność(czasami warto dopuścić pewną ilość błędów niektóre dane są 'wyjątkami')

dodatkowo KNN (K najbliższych sąsiadów) jeden z algorytmów regresji nieparametrycznej używanych w statystyce do prognozowania wartości pewnej zmiennej losowej. Może również być używany do klasyfikacji. k-NN może być używane do prognozowania popytu poprzez identyfikowanie podobnych wzorców historycznego popytu do obecnej sytuacji. używamy dane historyczne dotyczących popytu jako zestawu treningowego. Każdy punkt danych reprezentuje przeszły wzorec popytu i zawiera cechy takie jak pora roku, promocje, itp. Gdy chcesz prognozować przyszły popyt, obliczasz podobieństwo (odległość) między obecną sytuacją a k-najbliższymi historycznymi wzorcami popytu.

Drzewo decyzyjne, jest intuicyjnym narzędziem do prognozowania, które pozwala na interpretację wyników i jest często używane w analizie danych i uczeniu maszynowym. są często stosowane w prognozowaniu popytu, aby uwzględnić hierarchiczną strukturę czynników wpływających na popyt. Jak to działa: Buduje drzewo decyzyjne na podstawie danych historycznych, gdzie popyt jest zmienną docelową, a różne czynniki (np. cena, wydatki, sezonowość) są cechami wejściowymi. Drzewo jest budowane poprzez rekurencyjne dzielenie danych na podzbiory na podstawie najważniejszej cechy, aż zostaną spełnione kryteria zatrzymania.

Modele ML dzielą dane na zbiór uczący i testowy, dla każdego z nich bardzo ważna jest jakość danych które się wprowadza, odpowiednie oznaczanie cech, żaden z nich nie będzie dobrze działał jeśli zostaną wprowadzone do uczenia dane złej jakości. Dobór danych do uczenia ma bezpośredni wpływ na działanie każdego algorytmu ML.

Np. SVM bardzo dobrze radzi z wieloma czynnikami. Zdolność do radzenia sobie z danymi o wielu wymiarach: SVM są skuteczne w przekształcaniu danych do przestrzeni o wysokiej liczbie wymiarów. Efektywnie radzi sobie z danymi nieliniowymi: SVM mają wbudowaną zdolność do radzenia sobie z danymi nieliniowymi, dzięki zastosowaniu funkcji jądrowych, które pozwalają na transformację danych do przestrzeni o wyższej wymiarowości, gdzie mogą być liniowo separowalne. RNN i NN jak i SVM dobrze radzą sobie ze skomplikowanymi danymi, nieliniowymi zależnościami. Nieliniowość jest jednym

z kluczowych atutów modeli maszynowego uczenia nad tradycyjnymi modelami opartymi o szeregi czasowe. Oto dlaczego nieliniowość jest główną zaletą tych modeli: Zdolność do modelowania skomplikowanych zależności: W rzeczywistych problemach, zależności między danymi często są skomplikowane i nieliniowe. Zdolność do rozwiązywania problemów klasyfikacji nieliniowej: W zadaniach klasyfikacji, gdzie celem jest podział danych na różne klasy, istnieją sytuacje, w których granica decyzyjna między klasami nie jest liniowa. Zdolność do wydobywania nieliniowych wzorców w danych: W przypadku analizy danych, często istnieją nieliniowe wzorce lub zależności, które mogą być trudne do wykrycia za pomocą tradycyjnych modeli liniowych. Elastyczność w dostosowywaniu się do danych: Modele nieliniowe pozwalają na bardziej elastyczne dopasowanie do różnych rodzajów danych

Wsteczna propagacja błędu w czasie, pozwala na uczenie się wzorców na dowolnej głębokości w szeregu czasowym. Sama wstępna propagacja pomaga na bieżąco dopasowywać model: Wsteczna propagacja błędu w czasie (ang. Backpropagation Through Time, BPTT) to technika uczenia sieci RNN poprzez dostosowywanie wag na podstawie błędów prognoz w kolejnych krokach czasowych. Pozwala to na aktualizację wag w sieci, aby minimalizować błąd prognozowany przez model.

Dodatkowo tradycyjne metody często mają błąd tzw. efekt bata . Efekt "bata" (ang. "bullwhip effect") jest zjawiskiem występującym w łańcuchu dostaw i zarządzaniu zapasami, które polega na zwiększającej się zmienności i niestabilności zamówień w miarę przesuwania się w górę łańcucha dostaw od ostatecznego punktu sprzedaży do dostawcy surowców. To zjawisko może prowadzić do nadmiernego wzrostu kosztów i nadmiar zapasów, co jest niekorzystne dla efektywności łańcucha dostaw i rentowności firmy. Główne przyczyny efektu "bata" obejmują:

- Nadmierną zmienność zamówień: Zamówienia przekazywane przez kolejne ogniwa łańcucha dostaw mogą być podatne na oscylacje, gdzie zamówienia na jednym etapie mogą znacznie różnić się od rzeczywistego popytu na końcu łańcucha dostaw.

- Opóźnienia w komunikacji: Informacje o rzeczywistym popycie i zamówieniach mogą być przekazywane z opóźnieniem między różnymi uczestnikami łańcucha dostaw, co powoduje nieaktualne i niestabilne prognozy.

- Zwiększanie zamówień bez odpowiedniej przyczyny: W odpowiedzi na niestabilność i oscylacje w zamówieniach, uczestnicy łańcucha dostaw mogą zwiększać zamówienia, aby zabezpieczyć się przed niedoborami, co dalej pogarsza efekt "bata".

Z efektem 'bata' ML radzi sobie znacznie lepiej od tradycyjnych metod ponieważ:

- Modele ML, takie jak SVM, RNN i ANN, mają zdolność do wykrywania bardziej skomplikowanych wzorców w danych i uwzględniania nieliniowych zależności. Mogą bardziej precyzyjnie modelować zmiany w popycie.

- Lepsza adaptacja do danych czasowych: Modele są przydatne do prognozowania

danych czasowych, takich jak historia zamówień, ponieważ uwzględniają sezonowość i zmienne warunki

-Automatyczna adaptacja: Modele ML mogą być trenowane i dostosowywane do zmieniających się warunków rynkowych i danych, co pozwala na lepsze dostosowanie się do dynamicznych sytuacji.

-Integracja z danymi w czasie rzeczywistym: Modele ML mogą być dostosowywane do analizy danych w czasie rzeczywistym, co pozwala na reakcję na zmiany w czasie rzeczywistym.

Dodatkowym ważnym aspektem jest aspekt jest w jakiej prognozie krótko czy długoterminowej widac bardziej zalety uczenia maszynowego. Zdecydowanie w krótkoterminowym. Krótkoterminowe prognozy (na kilka dni, tygodni lub kilka miesięcy).Uczenie maszynowe (ML) może być bardziej przydatne w krótkoterminowych prognozach, gdy mamy do czynienia z nieliniowymi zależnościami i dynamicznymi zmianami w danych. Modele ML mogą lepiej radzić sobie z przewidywaniem krótkoterminowych fluktuacji popytu, które mogą być wynikiem sezonowości, trendów rynkowych lub innych skomplikowanych czynników. Długoterminowym okresie zmiany sa bardziej przewidywalne i częściej mozna uzywac tradycyjnych metod związanych z szeregami czasowymi.Tradycyjne metody mogą nadal być skuteczne w długoterminowych prognozach, zwłaszcza w przypadku stabilnych czynników wpływających na popyt. Niemniej gdy przychodzi do długoterminowej prognozy i podjecia decyzji jaka metode uzyc to wg. Katherine Gail Nowadly and Sohyun Jung z Massachusetts Institue of Technology najczesciej stosowane sa SVM, ANN[30]

[31][32][33][15][34][35][21][36][37][30]

Dla oceny jakosci prognozy popytu czesto stosuje sie MAPE i RMSE. MAPE (Mean Absolute Percentage Error) to popularna metryka używana do oceny jakości prognoz w problemach prognozowania popytu lub innych zadań regresji. MAPE pozwala na ocenę dokładności prognoz w sposób procentowy, co jest szczególnie przydatne do porównywania prognoz między różnymi zestawami danych lub modelami.

$$\left(\frac{1}{n} \sum_{i=1}^n \frac{|aktualny_i - prognozowany_i|}{|aktualny_i|} \right) \cdot 100$$

MAPE [15]

RMSE to skrót od "Root Mean Square Error", co w tłumaczeniu na język polski oznacza "Pierwiastek błędu średniokwadratowego". Jest to miara błędu, która jest często używana w statystyce i analizie danych do oceny dokładności modelu lub prognoz w stosunku do rzeczywistych danych.

Różnica między nimi jest w 'kwadracie' potęgowanie w RMSE skutkuje ze jest bardziej wrażliwy na duże błędy niż MAPE.

wyrażenie matematyczne dla RMSE wygląda następująco:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2}$$

n to liczba punktów danych w zestawie testowym

Y_i to rzeczywista wartość w i -tym punkcie danych

\hat{Y}_i to wartość prognozowana przez model w i -tym punkcie danych

2.3.2. Optymalizacja zapasów

Wprowadzenie Optymalizacja zapasów (zarządzanie zapasami) polega na skutecznym zarządzaniu ilościami produktów przechowywanych w magazynach. Głównym celem jest zminimalizowanie kosztów magazynowania, jednocześnie zapewniając wystarczający poziom dostępności produktów, aby zaspokoić popyt klientów. Tradycyjne metody zarządzania zapasami, takie jak metoda ABC czy metoda EOQ (Economic Order Quantity), są stosunkowo statyczne i nie uwzględniają zmienności popytu i innych czynników. Samo zarządzanie zapasami skupia się na 4 kwestiach: Koncentruje się na czterech zasadniczych kwestiach:

- ile jednostek należy zamówić (lub wyprodukować) w danym czasie,
- kiedy należy złożyć zamówienie,
- które składniki zapasów wymagają szczególnej uwagi,
- czy można zabezpieczyć się przed wzrostem kosztów zapasów. Główne metody

stosowane to:

- Reguła 80/20 - na jej bazie została stworzona metodyka ABC, dzieli zapasy pod kątem wartości. Skoro 20 procent pozycji stanowi 80procent całkowitej wartości zapasu, to sugeruje odmienne podejście do sterowania zapasem tych 20procent pozycji, do procesu wyboru dostawców oraz do obsługi dostaw.

- Metoda ABC polegająca na podziale zapasów na trzy grupy. Podział ten oparty jest na założeniu, że zapasy, które ilościowo stanowią duży udział w zapasach ogółem, lecz mały pod względem wartościowym. I odwrotnie: są też takie zapasy, których udział wartościowy jest duży, a mały ilościowo. Dzieli materiały (lub wytwarzane produkty) na ważne, mniej ważne i nieważne.

- Metoda XYZ(metoda ilościowa) jest uzupełnieniem metody ABC, w której zapasy rozpatruje się z punktu widzenia regularności zapotrzebowania i dokładności prognozowania. Klasa X zawiera asortymenty, na które występuje regularne zapotrzebowanie (popyt). Klasa Y stanowi asortyment, na który występują sezonowe wahania w zapotrzebowaniu. Klasa Z składa się z asortymentu, na który występuje sporadyczne (okazjonalne) zapotrzebowanie.

- Model ekonomicznej (optymalnej) wielkości zamówienia - EOQ (Economic Order Quantity) jest najbardziej rozpowszechnioną koncepcją wykorzystywaną w zarządzaniu zapasami materiałów i towarów. Wraz ze wzrostem wielkości zamówienia wzrasta poziom przeciętnych zapasów, a to z kolei powoduje spadek kosztów tworzenia i wzrost kosztów utrzymania zapasów. Jeżeli natomiast częstotliwość zamówień się zwiększy, to wielkość przeciętnych zapasów spadnie, zmniejszą się także koszty utrzymania, a wzrosną koszty tworzenia zapasów.

- Metoda "dokładnie na czas"(just- in- time JIT) polega na tym że, system gospodarki zapasami, w którym niezbędne materiały wpływają dokładnie wtedy, kiedy są potrzebne, bez zakłóceń procesu produkcji, co pomaga organizacji kontrolować zapasy surowców,

ograniczając zapotrzebowanie na powierzchnię magazynową. System ten zmniejsza zakres niezbędnych nakładów organizacji na przestrzeń magazynową potrzebna do przechowywania materiałów i na same materiały. Polega ona na również na zamawianiu materiałów i części, w mniejszych partiach, zmniejszając tym samym nakłady na przestrzeń magazynową oraz nakłady zamrożone w samych zapasach.

Koszty zapasów możemy podzielić na 3 części:

- koszty tworzenia zapasów(ang. Ordering Costs):

Są to koszty związane z procesem zamawiania i tworzenia nowych zapasów. Obejmuje to koszty związane z przygotowywaniem zamówień(koszty procesów informacyjnych), kontrolowaniem poziomu zapasów, negocjacjami z dostawcami(Koszty informacyjne), kosztami przewozu, a także kosztami związanymi z ewentualnym przyjęciem i magazynowaniem nowych dostaw.

- koszty utrzymania zapasów,

To koszty związane z przechowywaniem i utrzymaniem zapasów w magazynie lub na innych powierzchniach przechowalniczych.Składają się z następujących elementów: koszty kapitałowe: inwestycje w zapasy, koszty magazynowania: magazyny fabryczne, magazyny publiczne, magazyny dzierżawne, magazyny własne, koszty obsługi zapasów: ubezpieczenie, podatki, koszty ryzyka: utrata wartości, uszkodzenia, ubytki, koszty zmiany lokalizacji.

-koszty niedoboru zapasów.

koszty przestoju produkcyjnych, To koszty związane z brakiem zapasów w momencie, gdy są one potrzebne ,koszty utraconej sprzedaży mogą obejmować straty z tytułu utraty klientów, spadku sprzedaży oraz koszty utraconej reputacji a także ewentualne koszty pilnych zamówień lub dostaw w celu zaspokojenia brakujących zapasów..

[38][39][40]

Wykorzystanie uczenia maszynowego w optymalizacji zapasów 'stare' metody są statyczne nie uwzględniają zmian, opierają się na max kilku cechach (zazwyczaj 2 cechy: cena, zapotrzebowanie) jakiegokolwiek zmiany wymagają dużej pracy ludzkiej, nie dostosowują się do zmian, nie są elastyczne, nie radzą sobie z dużą ilością danych, . ML ten sam algorytm może pracować zarówno dla małej jak i dużej firmy dla każdego produktu, pod produktu, półproduktu jak i rodzin produktowych (tam gdzie armia ludzi w biurze i magazynie jest potrzebna do obsługi używając tradycyjnych metod). Tradycyjne metody były wykorzystywane przez kilkadziesiąt lat ale możliwości jakie daje maszynowe uczenie pozwala na usprawnienie pracy

Maszynowe Uczenie oferuje:

Dopasowanie do zmiennego popytu: W przeciwieństwie do tradycyjnych metod, które zakładają stałe lub statystyczne wartości (np. stałą wielkość zamówienia w EOQ), algorytmy uczenia maszynowego są w stanie dostosowywać się do zmiennej natury popytu.

Mogą analizować historyczne dane sprzedaży i innych czynników wpływających na popyt, co pozwala na generowanie dokładniejszych prognoz

Real-time reakcje: Uczenie maszynowe pozwala na monitorowanie i dostosowywanie poziomów zapasów w czasie rzeczywistym. Gdy zmieniają się warunki rynkowe, takie jak sezonowość, trendy konsumenckie lub zmiany w dostawach, algorytmy ML mogą reagować szybciej niż statyczne metody.

Automatyzacja zamawiania ,Sztuczna inteligencja zapewnia przewagę w postaci automatycznego ponownego zamawiania, czyli funkcji, która może znacznie poprawić efektywność zapasów. Na podstawie przewidywania trendów sprzedaży i aktualnych poziomów zapasów sztuczna inteligencja może we właściwym czasie uruchamiać automatyczne zamówienia uzupełniające, zapewniając w ten sposób optymalny poziom zapasów.

Analiza wielu czynników: ML może uwzględniać wiele czynników jednocześnie. Oprócz historii sprzedaży może analizować dane meteorologiczne, trendy konsumenckie, promocje , zmiany cen i wiele innych. Dzięki temu może lepiej zrozumieć, co wpływa na popyt i dostosować się do tych zmian.

Zwiększona precyzja: ML może dostarczyć bardziej precyzyjne prognozy popytu, co pomaga w uniknięciu nadmiarowych zapasów (które generują koszty magazynowania) lub braków w dostawach (co wpływa na obsługę klienta i straty finansowe).

Wykrywanie anomalii: Algorytmy uczenia maszynowego są również zdolne do wykrywania anomalii w danych magazynowych, co pozwala na szybkie reagowanie na nieprawidłowości w poziomach zapasów.

Personalizacja: ML może dostosowywać strategie zarządzania zapasami do konkretnych rodzajów produktów, co pozwala na bardziej efektywne podejście w zależności od charakterystyki produktów. [41][42][43][44]

W literaturze spotykam poniższe algorytmy maszynowego uczenia stosowane do optymalizacji zapasów: SVM, SVR, RNN, ANN, drzewa decyzyjne, Q-Learning. większość z nich została już skrótowo opisana w poprzednim podrozdziale. Tutaj na uwagę zasługuje jeszcze nie opisywany SVR i Q-Learning

SVR (Support Vector Regression) to algorytm uczenia maszynowego wykorzystywany do rozwiązywania problemów regresji, czyli przewidywania wartości numerycznych na podstawie danych treningowych. Działa on na podobnej zasadzie co SVM (Support Vector Machine) w przypadku klasyfikacji, ale jest dostosowany do problemów regresji zaś SVM klasyfikuje dane i je dzieli. Regresja to pojęcie w statystyce i analizie danych, które odnosi się do procesu modelowania zależności między jedną lub wieloma zmiennymi niezależnymi (czynnikami) a zmienną zależną (wartością, którą chcemy przewidzieć). Głównym celem regresji jest znalezienie matematycznego modelu lub funkcji, która opisuje, jak zmienne niezależne wpływają na zmienną zależną.

Q-Learning to algorytm uczenia maszynowego, który jest często wykorzystywany do uczenia się optymalnych strategii w problemach decyzyjnych z uczeniem ze wzmocnie-

niem. Ogólna idea Q-Learning polega na uczeniu agenta, jakie akcje podejmować w danej sytuacji, aby maksymalizować sumę nagród na przestrzeni wielu kroków czasowych. Algorytm ten jest często używany w kontekście gier i problemów, w których agent interaktywnie działa w pewnym środowisku. Oto główne kroki algorytmu Q-Learning:

- Inicjalizacja tabeli Q: Tworzona jest tabela (lub macierz) Q, która przechowuje wartości Q dla wszystkich dostępnych stanów i akcji.

- Wybór akcji: Agent rozpoczyna w pewnym stanie środowiska. Na podstawie wartości Q wybierana jest akcja, którą agent chce podjąć. Może to być akcja wybrana w sposób zachłanny (na podstawie maksymalnej wartości Q)

- Wykonanie akcji: Agent wykonuje wybraną akcję i przechodzi do nowego stanu środowiska.

- Obserwacja nagrody i nowego stanu: Agent obserwuje, ile nagrody otrzymuje za podjętą akcję oraz jaki jest nowy stan środowiska, do którego przeszedł.

- Aktualizacja wartości Q: Wartość Q dla danej pary stanu i akcji jest aktualizowana na podstawie otrzymanej nagrody i przewidywanej przyszłej wartości Q. Istnieje wiele czynników, które sprawiają, że Q-learning ma kluczowe znaczenie:

Dlaczego używamy q-learning? Po pierwsze, bez wyraźnego programowania komputery mogą uczyć się na nowych ustawieniach i dostosowywać się do nich. Nazywa się to Q-learningiem. Należałoby napisać wyraźne instrukcje dla każdej istotnej okoliczności, z jaką może spotkać się komputer w tradycyjnym programowaniu. Komputer jest bardziej wszechstronny i dostosowuje się do nowych scenariuszy dzięki Q-learningowi, który pozwala mu na samodzielną naukę metodą prób i błędów. Po drugie, różnorodne procesy decyzyjne, w tym te w robotyce, teorii gier i finansach, można zoptymalizować za pomocą Q-learningu. Q-learning może pomóc komputerom w dokonywaniu bardziej przydatnych ocen w skomplikowanych kontekstach poprzez identyfikację najlepszego sposobu działania lub zbioru działań, które maksymalizują długoterminową nagrodę.

Odnosnie zarządzania zapasami, q-learning jest w stanie podejmować decyzje co zamówić? kiedy? ile? po jakiej cenie? jest w stanie podejmować optymalne decyzje w środowisku, które jest dynamiczne, ma bardzo dużo danych, dzięki permanentnej eksploracji ten model jest w stanie osiągnąć znacznie więcej niż człowiek z tradycyjnymi modelami. Można bez problemu wprowadzić otoczenie dla agenta, gdzie będzie koszt zamawiania, koszt zakupu, koszt trzymania materiału, przychód ze sprzedaży, koszt braku materiału na stanie itp...

[44][45][46][47]

2.3.3. Planowanie produkcji

Planowanie produkcji to uzgodnienie asortymentu oraz ilości wyrobów gotowych, które są przewidziane do wytworzenia, a także rozłożenie ich produkcji w czasie, tak aby zrealizować plan sprzedaży i osiągnąć zakładany zysk i aby sprostać zapotrzebowaniu rynku. Jest to istotny element zarządzania łańcuchem dostaw oraz efektywnego wykorzystywania zasobów produkcyjnych. Planowanie dotyczy zarówno aspektów technicznych, jak i organizacyjnych, ekonomicznych i zarządzania. Ważne parametry to specyfika wyrobów, terminy rozpoczęcia i zakończenia procesu produkcyjnego oraz chronologiczne czynności w procesie wytwarzania. Planowanie produkcji może być na rok, uwzględniając zdolności produkcyjne i zasoby, oraz wsteczne lub do przodu. Główny harmonogram produkcji ma na celu efektywne wykorzystanie zasobów, utrzymanie poziomu obsługi klientów i minimalizację kosztów. Istnieją różne warianty tworzenia harmonogramu, a planowanie produkcji może być międzywydziałowe, wewnątrzwydziałowe lub ogólnozakładowe. Planowanie produkcji może być operacyjne, taktyczne lub strategiczne, zależnie od horyzontu czasowego i hierarchii w procesie planowania. Planowanie produkcji obejmuje planowanie procesów, surowców i zasobów w celu wytworzenia towarów dla konsumentów w określonych ramach czasowych. Harmonogram produkcji określa, kto i kiedy będzie przeprowadzał operacje.

Główne cele planowania produkcji to:

1. Ciągłość przepływu produkcji – jednym z głównych celów planowania i kontroli produkcji jest ciągły przepływ produkcji przez zakład produkcyjny. Ciągłe doskonalenie przepływu i procesów to jedna z integralnych koncepcji Lean Manufacturing, Six Sigma i ogólnego usprawnienia produkcji. Planowanie i kontrola produkcji będzie próbą osiągnięcia płynnego i ciągłego procesu produkcyjnego poprzez wyeliminowanie wąskich gardeł i marnotrawstwa z zakładu, co pozwoli przenieść produkcję na wyższy poziom pod względem produktywności.

2. Planowane zasobów — element planowania produkcji i kontroli produkcji jest niezbędny, aby mieć pewność, że zakład produkcyjny ma to, czego potrzebuje, we właściwej ilości i we właściwym czasie. Posiadanie planu produkcji gwarantuje, że masz wystarczającą ilość materiałów, maszyn, narzędzi, sprzętu i siły roboczej do wykonania pracy.

3. Optymalny poziom zapasów — zapasy są często uważane za marnotrawstwo w operacjach produkcyjnych. Dzieje się tak dlatego, że koszty związane z zapasami stanowią jedno z największych kosztów ponoszonych w przypadku wielu operacji. Dlatego też należy dążyć do maksymalnego zmniejszenia poziomu zapasów, aby odpowiednio obniżyć koszty w swojej działalności. Planowanie i kontrola produkcji planuje oraz realizuje procesy, które umożliwiają spójny przepływ produkcji, aby ostatecznie zbliżyć się do produkcji JIT (dokładnie na czas) i zmniejszyć poziom zapasów w operacjach produkcyjnych.

4. Zwiększona produktywność – zwiększona produktywność to kolejny istotny cel w ramach planowania i kontroli produkcji. Planowanie i kontrola produkcji ostatecznie ma na celu zwiększenie produktywności poprzez poprawę wydajności, a jednocześnie jest ekonomiczne. Zwiększoną produktywność udaje się osiągnąć poprzez optymalizację wykorzystania istniejących zasobów produkcyjnych i zasobów pracy, przy jednoczesnej eliminacji marnotrawstwa/psucia materiałów.

5. Zadowolenie klienta - Wszystkie firmy chcą zwiększać zadowolenie klientów. Dzieje się tak dlatego, że popyt jest napędzany przez klientów, a zadowoleni klienci oznaczają więcej transakcji w przyszłości. Posiadanie wadliwych produktów i opóźnienia w dniach wysyłki będą tylko szkodliwe dla firmy. Planowanie i kontrola produkcji koncentruje się przede wszystkim na zapewnieniu, że proces produkcyjny skutecznie i terminowo realizuje zamówienia oraz skróceniu czasu realizacji. Dzięki temu zamówienia klientów zostaną zrealizowane terminowo i w krótkim czasie, co wpłynie na poprawę zadowolenia klientów z Twojej firmy.

Podziały planowania produkcji:

1. Planowanie strategiczne produkcji: To długoterminowe planowanie, które obejmuje okres od kilku lat do kilkunastu lat. Służy osiągnięciu celów strategicznych oraz zrealizowaniu misji przedsiębiorstwa. Powinno uwzględniać zmiany w otoczeniu rynkowym. Jego podstawą są prognozy ekonomiczne, badania marketingowe, a także zapotrzebowanie na wyrób, ustala ogólne kierunki rozwoju produkcji i dostosowanie jej do strategii przedsiębiorstwa. W tym rodzaju planowania uwzględnia się m.in. inwestycje w nowe zakłady produkcyjne, rozwijanie nowych produktów i technologii.

2. Planowanie taktyczne produkcji: To planowanie na średni okres czasu, zazwyczaj obejmuje okres od kilku miesięcy do kilku lat. Obejmuje przygotowanie produkcji (m.in. plan zapotrzebowania na niezbędne zasoby, zarządzanie zapasami, utrzymanie maszyn, sterowanie jakością wyrobów), marketing, dystrybucję wyrobów, badanie i rozwój wyrobu oraz plany przepływów środków finansowych. Celem jest dostosowanie produkcji do zmieniającego się popytu, zachowując przy tym odpowiedni poziom zasobów i elastyczność w produkcji. W ramach tego planowania tworzy się tzw. plany produkcyjne, które określają harmonogramy produkcji na podstawie prognoz popytu.

3. Planowanie operacyjne produkcji: To planowanie krótkoterminowe, obejmuje najbliższe dni, tygodnie lub miesiące. Odnosi się do działań jednorazowych, które nazywamy projektami. Najczęściej obejmują przygotowanie produkcji oraz działania powtarzalne, które dotyczą procesów wytwarzania wyrobów. Celem jest precyzyjne ustalenie harmonogramu produkcji na podstawie aktualnych danych o stanie magazynów, dostawach surowców i aktualnym popycie. W ramach tego planowania tworzy się tzw. harmonogramy produkcyjne, które określają, co i kiedy ma być produkowane.

4. Planowanie reaktywne: To rodzaj planowania, które występuje w przypadku nieoczekiwanych zakłóceń w procesie produkcji, takich jak awarie maszyn, opóźnienia dostaw,

czy zmiany w zamówieniach klientów. Celem jest szybka reakcja na te zakłócenia i minimalizacja negatywnych skutków dla produkcji.

Podział planowania produkcji ze względu na typ produkcji:

1. Planowanie oparte na zadaniu ,jednostkowa (custom-made)

Produkcja oparta na zadaniu lub projekcie koncentruje się na wytwarzaniu pojedynczego produktu i jest obsługiwana przez jednego pracownika lub grupę osób (każdy produkt jest wykonywany na zamówienie klienta, zazwyczaj w pojedynczych egzemplarzach lub w niewielkich seriach), w przypadku zadań na małą skalę, które wymagają bardzo mało specjalistycznego sprzętu, jest stosunkowo łatwe do wykonania. Rodzaje zadań objętych tego typu planowaniem produkcji mogą dotyczyć małej skali, np. tworzenia biżuterii na zamówienie. Większe, bardziej złożone projekty produkcyjne, takie jak budowa domów na zamówienie, również należą do tej kategorii. Pozwala to na wykonanie produktów według życzeń klienta i zazwyczaj może zostać włączone w dowolnym momencie procesu produkcyjnego, nie zakłócając jego przebiegu. Planowanie produkcji jednostkowej jest bardzo elastyczne i opiera się na dostosowaniu do indywidualnych potrzeb klienta. Procesy produkcyjne są unikalne dla każdego produktu.

2. Produkcja seryjna (batch production)

Produkcja seryjna ma zastosowanie, gdy elementy są produkowane w grupach, a nie indywidualnie lub w ramach produkcji ciągłej. Produkty są wytwarzane w partiach lub seriach o stałej wielkości. Na przykład ciastka są produkowane partiami, co oznacza, że każdy etap produkcji w danej partii ciastek odbywa się w tym samym czasie. Zaczyna się od odmierzenia składników całej porcji, następnie zmieszasz je ze sobą, a na koniec upieczesz je razem, tak aby cały proces produkcji partii ciasteczek rozpoczynał się i kończył w tym samym czasie.

Wyzwaniem, , jest uwzględnienie ograniczeń na każdym etapie operacji, aby zapewnić maksymalizację wydajności zasobów bez przekraczania maksymalnego dozwolonego limitu. Na przykład, jeśli Twój mikser do ciasta zmieści partię 100 ciastek, ale możesz upiec tylko 300 na raz, możesz napotkać wąskie gardła w produkcji.

Planowanie produkcji seryjnej ma na celu zoptymalizowanie procesów produkcyjnych dla konkretnej partii, minimalizując zmiany w konfiguracji maszyn i narzędzi.

3. Produkcja przepływowej (flow production)

Produkcja przepływowa to metoda sterowana popytem, charakteryzująca się ciągłym przepływem jednostek przez linię produkcyjną. Produkcja przepływowa (flow production) obejmuje procesy produkcyjne, w których produkty są wytwarzane w ciągłym przepływie, bez przerw i z minimalnymi zmianami w konfiguracji maszyn , narzędzi . Ten typ produkcji jest często stosowany w sektorach takich jak przemysł motoryzacyjny, produkcja elektroniki ,przemysł spożywczy lub w produkcji telewizorów i sprzętu AGD, gdzie produkt wytwarzany jest w drodze szeregu zbiorowych operacji, podczas których materiały przechodzą z jednego etapu do drugiego bez opóźnień i przerw.

Kluczowe cechy produkcji przepływowej obejmują:

Ciągłość produkcji: Procesy produkcyjne działają nieprzerwanie, aby utrzymać stały przepływ produktów przez linię produkcyjną.

Standaryzacja: Produkty są zazwyczaj standardowe i podlegają minimalnym zmianom w trakcie produkcji.

Automatyzacja: W produkcji przepływowej stosuje się automatyzację w dużym stopniu, aby zminimalizować udział pracy ręcznej i zwiększyć wydajność.

Optymalizacja przepływu: Planowanie produkcji przepływowej skupia się na optymalizacji sekwencji operacji produkcyjnych, aby zapewnić płynny przepływ produktów.

Wysoka wydajność: Produkcja przepływowa ma na celu osiągnięcie jak najwyższej wydajności i minimalizację kosztów jednostkowych.

Korzyści płynące z metody przepływowej produkcji polegają na tym, że producenci mogą zminimalizować liczbę produktów w toku i wyrobów gotowych przechowywanych w zapasach, obniżyć koszty i skrócić czas realizacji produkcji.

4. Metoda masowej produkcji

Produkcja masowa jest bardzo podobna do produkcji przepływowej. Technika ta jest bardzo korzystna przy produkcji dużej liczby tych samych przedmiotów w krótkim czasie. Produkty są wytwarzane w dużej skali, zazwyczaj w stałych seriach.

Ten rodzaj produkcji jest zwykle zautomatyzowany, co zmniejsza koszty pracy potrzebnej do produkcji. Niektóre zakłady produkcyjne posiadają linie montażowe dedykowane dla określonego rodzaju przedmiotu, co skraca wymagany czas przeobrażenia i zwiększa ogólną wydajność produkcji. Pozwala to producentom zwiększyć zyski, ponieważ koszty produkcji są znacznie obniżone.

Planowanie produkcji masowej ma na celu osiągnięcie maksymalnej wydajności i minimalizację kosztów jednostkowych.

5. Produkcja ciągła lub procesowa (continuous production): W produkcji ciągłej procesy produkcyjne działają przez całą dobę bez przerwy, rodzaj procesu ciągłego podobnego do produkcji masowej i produkcji przepływowej, ale charakteryzuje się ciągłym przepływem materiałów przez linię produkcyjną. Ten typ produkcji jest często stosowany w branżach chemicznej, petrochemicznej i wytwarzania energii. Planowanie produkcji ciągłej ma na celu utrzymanie stałego przepływu produkcji i minimalizację przestojów. Zazwyczaj wyroby gotowe wytworzone w tego typu produkcji nie są liczone jako odrębne jednostki. Na przykład produkcja i przetwarzanie cieczy, gazów lub chemikaliów, gdy produkt jest wytwarzany w jednolitej i znormalizowanej kolejności.

Metoda Procesowa wykorzystuje specyficzne i wyrafinowane maszyny do przetwarzania materiałów na każdym etapie operacji. W tego typu produkcji nie ma miejsca na błędy, ponieważ zmiana jednego typu przedmiotu na inny będzie wymagała długiego okresu przeobrażenia. Często w wyniku tego rodzaju produkcji powstają produkty uboczne lub odpady.

6. Produkcja w linii montażowej (assembly line production): Ten typ produkcji opiera się na przemieszczaniu produktu przez linię montażową, gdzie każdy etap montażu jest wykonywany przez innego pracownika lub maszynę. Planowanie produkcji w linii montażowej ma na celu zoptymalizowanie sekwencji i czasu wykonywania poszczególnych operacji.

7. Produkcja projektowa (project production): W produkcji projektowej każdy produkt lub projekt jest unikalny, a procesy produkcyjne są dostosowywane do konkretnych projektów. Planowanie produkcji projektowej wymaga skomplikowanej koordynacji zasobów i harmonogramowania działań.

Podział ten wynika z różnic w rodzaju produktów, ilościach produkowanych jednostek, rodzaju procesów produkcyjnych i elastyczności produkcji. Planowanie produkcji musi być dostosowane do specyficznych wymagań każdego typu produkcji, aby osiągnąć efektywność i spełnić oczekiwania klientów.

Istnieje wiele innych odmian planowania produkcji, które koncentrują się na różnych aspektach procesu produkcyjnego. Oto kilka z nich:

A. Główny harmonogram produkcji (Master production schedule MPS)

Są to harmonogramy produkcji dla określonych towarów, które muszą być produkowane oddzielnie. Często są tworzone przez oprogramowanie, a następnie modyfikowane przez użytkowników. Plan ten określa ilościowo istotne procesy, części i inne zasoby w celu optymalizacji produkcji, identyfikacji wąskich gardeł oraz przewidywania potrzeb i gotowych towarów. Ponieważ MPS napędza dużą działalność fabryki, jego dokładność dramatycznie wpływa na rentowność. Typowe MPS są tworzone przez oprogramowanie z możliwością modyfikacji przez użytkownika. Zwykle jest to powiązane z produkcją, gdzie plan wskazuje, kiedy i ile będzie zapotrzebowanie na każdy produkt. Plan ten określa ilościowo istotne procesy, części i inne zasoby w celu optymalizacji produkcji, identyfikacji wąskich gardeł oraz przewidywania potrzeb i gotowych towarów. Harmonogramy nie obejmują wszystkich aspektów produkcji, a jedynie kluczowe elementy, które udowodniły swoją skuteczność kontrolną, takie jak prognozowany popyt, koszty produkcji, koszty zapasów, czas realizacji zamówienia, godziny pracy, wydajność, poziom zapasów, dostępne miejsce do przechowywania i dostawę części. Wybór tego, co modelować, różni się w zależności od firmy i fabryki. MPS przekłada zapotrzebowanie klientów (zamówienia sprzedaży) na plan budowy, wykorzystując zaplanowane zamówienia w prawdziwym środowisku planowania komponentów. Korzystanie z MPS pomaga uniknąć niedoborów, kosztownych przyspieszeń, planowania w ostatniej chwili i nieefektywnej alokacji zasobów. Jak działa mps?. Wykorzystując wiele zmiennych jako dane wejściowe, MPS wygeneruje zestaw wyników wykorzystywanych do podejmowania decyzji. Dane wejściowe mogą obejmować prognozowany popyt, koszty produkcji, wartość zapasów, potrzeby klientów, dostawy, wielkość partii, czas realizacji produkcji i wydajność. Dane wejściowe mogą być automatycznie generowane przez system ERP łączący dział sprzedaży z działem

produkcji. Wyniki mogą obejmować ilości, które mają zostać wyprodukowane, poziom zatrudnienia, dostępną ilość do obiecania i przewidywane dostępne saldo. Wyniki można wykorzystać do stworzenia harmonogramu planowania wymagań materiałowych (MRP).

Efektywny MPS :

Zapewni produkcji, planowaniu, zakupom i zarządzaniu informacje umożliwiające planowanie i kontrolowanie produkcji. Powiąże ogólne planowanie biznesowe i prognozowanie ze szczegółowymi operacjami. Umożliwi marketingowi podejmowanie uzasadnionych zobowiązań w zakresie dostaw do magazynów i klientów. Zwiększy wydajność i dokładność produkcji w firmie zapewni zgrubne planowanie wydajności.

B. Planowanie zapotrzebowania materiałowego (MRP)

MRP to system zarządzania zapasami, harmonogramowania i planowania produkcji. Dostępność surowców gwarantuje system MRP, który utrzymuje również wewnętrzne poziomy materiałów i produktów na możliwie najniższym poziomie.

Planowanie wymagań materiałowych (MRP) to system planowania, harmonogramowania i kontroli zapasów, używany do zarządzania procesami produkcyjnymi. Większość systemów MRP opiera się na oprogramowaniu. System MRP ma jednocześnie spełniać trzy cele:

- Upewnij się, że surowce są dostępne do produkcji, a produkty są dostępne do dostawy do klientów.

- Utrzymuj najniższy możliwy poziom materiałów i produktów

- Planuje działania produkcyjne, harmonogramy dostaw i działania zakupowe.

Do podstawowych funkcji systemu MRP zalicza się: kontrolę zapasów, przetwarzanie list materiałowych i elementarne planowanie. Służy do planowania działań związanych z produkcją, zakupami i dostawami. MRP jest narzędziem pozwalającym uporać się z tymi problemami. Zawiera odpowiedzi na kilka pytań: Jakie elementy są wymagane? Ile jest potrzebnych? Kiedy są potrzebne?... Dane, które należy wziąć pod uwagę, obejmują:

Tworzony element końcowy . Nazywa się to czasem niezależnym popytem lub poziomem „0” na BOM (bill of material zestawieniu materiałów).

Ile potrzeba teraz.

Kiedy ilości są wymagane, aby zaspokoić popyt.

Okres trwałości przechowywanych materiałów.

Stan zapasów. Ewidencja materiałów netto dostępnych do wykorzystania już w magazynie (od ręki) oraz materiałów na zamówienie u dostawców.

Zestawienia materiałów. Szczegóły dotyczące materiałów, komponentów i podzespołów wymaganych do wytworzenia każdego produktu.

Dane planistyczne. Obejmuje to wszystkie ograniczenia i wskazówki dotyczące produkcji takich przedmiotów, jak: trasa, standardy pracy i maszyn, standardy jakości i testowania, polecenia pull/work cell i push, techniki ustalania wielkości partii (tj. stała wielkość partii, partia za partią), procent złomowania i inne dane wejściowe.

dane wyjściowe:

Wyjścia

Istnieją dwa wyjścia i różne komunikaty/raporty:

Wynik 1 to „Zalecany harmonogram produkcji”. Określa szczegółowy harmonogram wymaganych dat rozpoczęcia i zakończenia, wraz z ilościami dla każdego etapu wyznaczania trasy i zestawienia materiałów wymaganych do zaspokojenia zapotrzebowania z głównego harmonogramu produkcji (MPS).

Wynik 2 to „Zalecany harmonogram zakupów”. Określa to zarówno daty, w których zakupione artykuły powinny zostać przyjęte do zakładu, jak i daty, w których powinny nastąpić zamówienia zakupu lub wydanie zamówienia zbiorczego, aby dopasować się do harmonogramów produkcji.

C. Planowanie zdolności produkcyjnych (capacity planning)

Jest to proces sprawdzania, jak dobrze firma jest przygotowana do radzenia sobie ze zmieniającymi się wymaganiami. To proces określania zdolności produkcyjnej potrzebnej organizacji, aby sprostać zmieniającemu się zapotrzebowaniu na jej produkty. W kontekście planowania wydajności zdolność projektowa to maksymalna ilość pracy, jaką organizacja lub osoba jest w stanie wykonać w danym okresie. Rozbieżność między możliwościami organizacji a wymaganiami jej klientów skutkuje nieefektywnością wynikającą z niedostatecznego wykorzystania zasobów lub niespełnionego zapotrzebowania klientów. Celem planowania zdolności produkcyjnych jest zminimalizowanie tej rozbieżności. Zapotrzebowanie na zdolność organizacji zmienia się w zależności od zmian w wielkości produkcji, takich jak zwiększanie lub zmniejszanie wielkości produkcji istniejącego produktu lub wytwarzanie nowych produktów. Lepsze wykorzystanie istniejącej wydajności można osiągnąć poprzez poprawę ogólnej efektywności sprzętu (OEE).

szerekie klasy planowania zdolności produkcyjnych obejmują strategię wiodącą, strategię opóźnień, strategię dopasowania i strategię dostosowania:

- Wiodącą strategią jest zwiększanie wydajności zgodnie z rosnącym popytem. Strategia wiodąca to strategia agresywna, której celem jest odciążenie klientów od konkurentów firmy poprzez poprawę poziomu usług i skrócenie czasu realizacji. To także strategia mająca na celu redukcję kosztów braków magazynowych

- Strategia opóźnienia odnosi się do dodawania mocy produkcyjnych dopiero wtedy, gdy organizacja pracuje z pełną wydajnością lub przekracza ją ze względu na wzrost popytu. Jest to strategia bardziej konserwatywna i przeciwna strategii wiodącej wydajności. Zmniejsza to ryzyko marnotrawstwa, ale może skutkować utratą potencjalnych klientów w wyniku wyczerpania zapasów lub niskiego poziomu usług

- Strategia dopasowania polega na zwiększaniu wydajności w małych ilościach w odpowiedzi na zmieniający się popyt na rynku. Jest to bardziej umiarkowana strategia.

D. Planowanie produkcji poziomowane (level production planning)

to rodzaj planowania produkcji, który koncentruje się na wytwarzaniu stałej wielkości

produkcji w danym okresie. Oznacza to, że przez cały okres produkcji wykorzystywana jest ta sama ilość surowców i zasobów, co zapewnia stałą i przewidywalną produkcję.

Celem tego rodzaju planowania jest osiągnięcie wysokiej efektywności i redukcja kosztów poprzez zapewnienie możliwie najbardziej efektywnego wykorzystania zasobów.

E. Planowanie odchudzone (lean production planning)

Lean Production Planning to rodzaj planowania produkcji, który koncentruje się na minimalizacji odpadów i optymalizacji wykorzystania zasobów. Ten rodzaj planowania kładzie nacisk na stosowanie małych partii i eliminację nadprodukcji. Celem odchudzonego planowania produkcji jest redukcja kosztów, zwiększenie wydajności i poprawa zadowolenia klientów.

F. Planowanie produkcji Kaizen

Planowanie produkcji Kaizen to rodzaj planowania produkcji, który koncentruje się na ciągłym doskonaleniu i optymalizacji procesów. Ten typ planowania kładzie nacisk na wykorzystanie danych i informacji zwrotnych w celu zidentyfikowania obszarów wymagających poprawy i wprowadzenia zmian, które zwiększą wydajność i obniżą koszty.

G Zwinne planowanie produkcji (agile planning)

Zwinne planowanie produkcji to rodzaj planowania produkcji, który koncentruje się na podejmowaniu szybkich i skutecznych decyzji. Ten rodzaj planowania kładzie nacisk na wykorzystanie danych i informacji zwrotnych w celu szybkiego podejmowania decyzji i szybkiego dostosowywania się do zmieniających się warunków.

Sam proces planowania produkcji wygląda następująco:

Opis procesu 1. Zdobycie informacji na temat zapotrzebowania (ile, jaki towar, kiedy i gdzie jest potrzebny)

2. Zdobycie informacji na temat stanów magazynowych (ile, jaki towar, gdzie jest)

3. Analiza pracochłonności i materiałochłonności oszacowanie materiałochłonności i pracochłonności określonych procesów produkcji.

4. Złożenie zamówienia na materiały i określenie potrzeb kadrowych

5. Hierarchizacja zleceń Efektem tej pracy jest informacja o kolejności realizacji zleceń.

6. Określenie przewidywanego czasu realizacji zamówienia Efektem pracy jest ustalenie planowych terminów realizacji zleceń

7. Sporządzenie szczegółowego planu produkcji

Efektem całej pracy jest końcowa uszczegółowiona postać planu produkcji

8. Aktualizacja bieżącego planu

9. Kontrola i monitorowanie produkcji

10. Informowanie o przebiegu procesu kierowników zaopatrzenia, sprzedaży

[48][49][50][51][52][53][54][40]

Wykorzystanie uczenia maszynowego w planowaniu produkcji Planowanie produkcji operuje w środowisku dynamicznym, zmiennym i bardzo złożonym. Zawsze są odstępstwa pomiędzy planowaniem produkcji i późniejszą realizacją. Te odchylenia wynikają z niepewności, np. niedokładne lub niewystarczające dane dotyczące planowania (np. jakość danych i dostępność), niewłaściwe systemy planowania i kontroli lub nieprzewidywalne zdarzenia. Dlatego planiści produkcji korzystają z buforów w postaci zapasów. Bufory jednak prowadzą do zwiększonego wysiłku w zakresie koordynacji i kontroli, zwiększając np. stan zapasów, czas realizacji. Ponadto stwierdzono, że niezawodność planów produkcyjnych, a tym samym jakość planowania (PQ) może spaść do 25 procent w ciągu pierwszych trzech dni po stworzeniu planu. Maszynowe uczenie właśnie musi umieć sobie radzić z wymienionymi problemami[55]

w literaturze fachowej najczęściej spotykane algorytmy/typy maszynowego uczenia stosowanych do planowania produkcji to: Uczenie przez wzmacnianie (Reinforcement Learning, RL) szczególnie q-learning (jeden z algorytmów uczenia przez wzmocnienie) jak i sieć neuronowa (NN neural networks), spotkałem się również z użyciem: regression tree, klastrowanie, drzewo decyzyjne, svm, bootstrap aggregated (bagged) tree, random forest (tutaj jako wsparcie w podejmowaniu decyzji dla poszczególnych elementów planowania), ensemble learning (uczenie zespołu). Ensemble learning to technika uczenia maszynowego, która łączy kilka podstawowych modeli w celu stworzenia jednego optymalnego modelu predykcyjnego gdzie tworzy się wiele modeli klasyfikacji lub regresji przy użyciu zbioru danych treningowych. Każdy model podstawowy może zostać utworzony przy użyciu różnych podziałów tego samego zestawu danych treningowych i tego samego algorytmu lub przy użyciu tego samego zestawu danych z różnymi algorytmami lub dowolnej innej metody następnie podejmowana jest decyzja jaki model wybrać za pomocą głosowania dla klasyfikacji i uśredniania dla regresji)

Uczenie przez wzmacnianie (Reinforcement Learning, RL) które jest dedykowane do niepewnego środowiska gdzie są rzeczy znane i nieznane. Środowisko jest zwykle określane w formie procesu decyzyjnego Markowa (MDP Markov Decision Process), ponieważ wiele algorytmów uczenia się przez wzmacnianie w tym kontekście wykorzystuje techniki programowania dynamicznego. Podstawowe uczenie się przez wzmacnianie jest modelowane jako proces decyzyjny Markowa: zbiór stanów środowiska i agenta, S , zbiór działań A agenta, prawdopodobieństwo przejścia (w chwili t) ze stanu s do stanu s' pod działaniem a , natychmiastowa nagroda po przejściu z s do s' z działaniem a .

Podsumowując Uczenie przez wzmacnianie składa się z elementów takich jak: - Agent: Agent to byt uczący się, który podejmuje decyzje w danym środowisku. (może tworzyć plan produkcji)

- Środowisko: Środowisko to kontekst, w którym agent działa i podejmuje decyzje. Środowisko może być rzeczywiste lub wirtualne. Środowisko dostarcza informacji zwrotnej w postaci nagród lub kary za działania agenta. To może być środowisko produkcyjne

-Akcje: Agent ma zestaw dostępnych akcji, które może podjąć w danym stanie środowiska. Celem agenta jest znalezienie strategii decyzyjnej (polityki), która określa, która akcja powinna być wybrana w odpowiedzi na stan środowiska. NP. może podejmować co i kiedy należy produkować

-Stany: Stany opisują aktualny stan środowiska. Mogą to być obserwacje dostępne dla agenta, które zawierają informacje o środowisku w danym momencie. Np. mogą to być informacje o dostępności materiałów, wykonaniu planu produkcji, awarii maszyn itp.

-Nagroda: Nagroda to liczba reprezentująca ocenę jakości działania agenta w danym stanie i po podjęciu danej akcji. Celem agenta jest maksymalizacja sumy nagród w dłuższej perspektywie czasowej. Np. możemy nagradzać za dobry plan produkcji np. taki który wykorzystał w pełni zdolności produkcyjne lub został w pełni wykonany (czyli był realny) zaspokoił zapotrzebowanie klienta itp.

-Polityka: Polityka to strategia decyzyjna agenta, która określa, jakie akcje należy podejmować w odpowiedzi na stany środowiska. Celem uczenia przez wzmacnianie jest znalezienie optymalnej polityki, która maksymalizuje nagrody. NP. tu można stosować różne strategie planistyczne np. level planning (równomierne planowanie produkcji)

w skrócie sam proces wygląda następująco:

1. Agent obserwuje aktualny stan środowiska.
2. Na podstawie obserwacji, agent używa swojej strategii decyzyjnej (polityki) do wyboru akcji.
3. Środowisko wykonuje wybraną akcję i przechodzi do nowego stanu.
4. Środowisko przekazuje nagrodę za wykonanie akcji.
5. Agent aktualizuje swoją strategię na podstawie otrzymanej informacji zwrotnej, aby w przyszłości podejmować lepsze decyzje.

Kilka słów odnośnie procesu decyzyjnego Markowa. Jest stochastycznym procesem sterowania w czasie dyskretnym. Zapewnia ramy matematyczne do modelowania podejmowania decyzji w sytuacjach, gdy wyniki są częściowo przypadkowe, a częściowo pod kontrolą decydenta. MDP są przydatne do badania problemów optymalizacyjnych rozwiązywanych za pomocą programowania dynamicznego. Nazwa MDP pochodzi od rosyjskiego matematyka Andrieja Markowa. Proces decyzyjny Markowa to 4-krotka, gdzie: (S, Z, P, R) S jest zbiorem stanów zwanym przestrzenią stanów, A to zbiór akcji zwany przestrzenią akcji, P to prawdopodobieństwo, że działanie A w stanie S w chwili T doprowadzi do stanu S' w chwili $t+1$, R to natychmiastowa nagroda (lub oczekiwana natychmiastowa nagroda) otrzymana po przejściu ze stanu S do stanu S' w wyniku działania A . Dlaczego uczenie przez wzmacnianie jest odpowiednie do planowania produkcji?

1. Dynamiczne i niepewne środowisko: Środowisko produkcyjne często jest dyna-

miczne i nieprzewidywalne. Procesy produkcyjne mogą być podatne na awarie maszyn, zmienne zapotrzebowanie na produkty, zmiany w dostawach surowców, i wiele innych czynników, które wpływają na efektywność i koszty produkcji. RL pozwala adaptować się do zmieniających się warunków i podejmować decyzje w czasie rzeczywistym.

2.Optymalizacja celów produkcyjnych: RL może być używane do optymalizacji celów produkcyjnych, takich jak maksymalizacja wydajności, minimalizacja kosztów lub optymalne zarządzanie zapasami. Agent RL może uczyć się optymalnej strategii decyzyjnej, która uwzględnia różne cele i ograniczenia produkcyjne.

3.Kompleksność problemu: Planowanie produkcji to zazwyczaj zadanie o dużym stopniu trudności, ze względu na wiele zmiennych i ograniczeń. RL pozwala na podejście do tych złożonych problemów i znalezienie rozwiązań w czasie rzeczywistym.

sa tez uzywane inne algorytmy ktore wspomagaja planiste produkcji (ale nie wykonuja calej pracy za niego) to wspomniane juz: Drzewo regresyjne (Regression Tree), Drzewo bagged (Bagged Tree), Las losowy (Random Forest) lub maszynowe uczenie polaczone z MPC (Model predictive control -model sterowania predykcyjnego). Odnosnie MPC (Model predictive control) jest to optymalna technika sterowania, w której obliczone działania sterujące minimalizują funkcję kosztu dla ograniczonego układu dynamicznego w skończonym, oddalającym się horyzoncie. W każdym kroku czasowym sterownik MPC otrzymuje lub szacuje aktualny stan . Następnie oblicza sekwencję działań kontrolnych, która minimalizuje koszty w horyzoncie czasowym, rozwiązując ograniczony problem optymalizacji. Modele stosowane w MPC mają generalnie na celu przedstawienie zachowania złożonych i prostych układów dynamicznych. Modele MPC przewidują zmianę zmiennych zależnych modelowanego systemu, która będzie spowodowana zmianami zmiennych niezależnych.

warto wspomniec ze również operacje logistyczne wspierające planowanie produkcji jak np. obsługa materialow (material handling) która to czynność polegająca na transporcie materiałów w zakładzie produkcyjnym, np. z magazynu do stanowiska pracy, lub ze stanowiska pracy do rampy załadunkowej lub pomiędzy budynkiem a pojazdem . MH odgrywa znaczącą rolę w płynnej i terminowej produkcji wyrobów gotowych . Tu również istnieje zmienne i dynamiczne srodowisko i mamy do czynienia ze zlozonym problemem czyli mamy srodowisko ktore pasuje dla RL czyli uczenie przez wzmacnianie.

[56][57][58][59][60][61][62][63][55][64][65][66][67][68][69]

2.4. Kluczowe algorytmy uczenia maszynowego dla łańcucha dostaw

W tym rozdziale przedstawione są kluczowe algorytmy uczenia maszynowego, które znajdują zastosowanie w zarządzaniu łańcuchem dostaw. Te algorytmy są wykorzystywane do rozwiązywania różnorodnych problemów związanych z planowaniem, prognozowaniem i optymalizacją w łańcuchach dostaw.

Wśród omawianych algorytmów znajdują się między innymi regresja liniowa, drzewa decyzyjne, sieci neuronowe, maszyny wektorów nośnych oraz algorytmy klastrowania. Dla każdego algorytmu przedstawiam także przykłady jego potencjalnych zastosowań w SCM. Celem tych rozdziałów jest zaprezentowanie podstawowych informacji na temat maszynowego uczenia i jego roli w zarządzaniu łańcuchem dostaw oraz dostarczenie wglądu w kluczowe algorytmy, które mogą być używane do rozwiązywania problemów związanych z SCM.

do tej pory wszystkie wymienione przez mnie modele i algorytmy maszynowego uczenia to:

Uczenie nadzorowane (Supervised Learning):

Uczenie nadzorowane polega na szkoleniu modelu za pomocą zbioru treningowego, w którym każda próbka jest opisana etykietą lub odpowiedzią. Celem jest nauczenie modelu przewidywania tych odpowiedzi dla nowych danych. Dwa główne obszary wykorzystujące uczenie nadzorowane to rozwiązywanie problemu regresji (przewidywanie wartości) i problem klasyfikacji (przewidywanie klas). Jako przykład regresji można wskazać prognozowanie wartości samochodu na podstawie jego modelu, marki, roku produkcji, przebiegu, pojemności silnika, rodzaju paliwa itd. Przykładem modelu klasyfikacji jest filtr spamu, czyli problem zakwalifikowania wiadomości jako spam czy nie. Obecnie ponad 90 procent modeli uczenia maszynowego wdrożonych do użycia to modele uczenia nadzorowanego.

Przykłady algorytmów uczenia nadzorowanego rozwiązujący problem regresji:

- Support vector regression
- Neural Networks (NN)
- Recurrent Neural Networks (RNN)
- K-nearest neighbors regression
- regression tree
- linear regression
- polynomial regression
- Generalized regression neural networks
- Classification and regression trees (CART)

Przykłady algorytmów uczenia nadzorowanego rozwiązujący problem klasyfikacji:

- Support Vector Machines (SVM)

- K-nearest neighbors (KNN)
- naive Bayes classifier
- random forest
- logistic regression
- decision tree

Uczenie nienadzorowane (Unsupervised Learning):

W uczeniu nienadzorowanym nie ma etykiet ani odpowiedzi w zbiorze treningowym. Głównym celem jest odkrywanie struktur lub wzorców w danych. Klastrowanie to przykład, w którym modele grupują dane na podstawie ich podobieństwa, podczas gdy redukcja wymiarowości pomaga zmniejszyć liczbę cech, zachowując istotne informacje. Mówiąc potocznie, posiadamy surowe dane które wrzucamy do modelu i zostawiamy algorytmowi całą pracę związaną ze znalezieniem powiązań między danymi (tzw. uczenie bez nauczyciela). Należą do nich m. in. algorytmy analizy skupień (ang. clustering) – inaczej klasteryzacja albo grupowanie. Przykładem problemu analizy skupień jest określenie grup podobnych użytkowników, robiących zakupy w sklepie internetowym albo wchodzących na serwis informacyjny.

- Klastrowanie (Clustering)
- Gaussian processes

Kilka najważniejszych algorytmów z podziałem na problemy, którymi się zajmują:

Analiza skupień (czyli podział na grupy o podobnych cechach tzw. klastry lub skupienia. Algorytm grupowania dzieli obserwacje bez żadnej wcześniejszej wiedzy jak docelowe grupy powinny wyglądać):

- metoda k-średnich
- hierarchiczna analiza skupień
- DBSCAN (density-based spatial clustering of applications with noise)

Wykrywanie anomalii i nowości (czyli przypadków znacząco różnych od pozostałych, przy założeniu że nie znamy które miałyby powodować te różnice):

- jedнокlasowa maszyna wektorów nośnych
- ograniczone maszyny Boltzmanna (nienadzorowane uczenie głębokie wyodrębia cechy które są • unikatowe dla każdej cyfry- wykrywa kształt linii i krzywych)

Wizualizacja i redukcja wymiarowości (redukcje wymiarów polegającego na zmniejszeniu liczby zmiennych żeby w jak najmniejszy stopniu wpłynęło na utratę informacji

lub autokodowania polegającego na wykrywaniu przez głębokie sieci neuronowe abstrakcyjnych cech przypadków widocznych na kodowanych obrazach kształtów):

- analiza głównych składowych (PCA – ang. principal component analysis)
- jądrowa analiza głównych składowych

Uczenie ze wzmocnieniem (Reinforcement Learning):

jeden z trzech głównych nurtów uczenia maszynowego, którego zadaniem jest interakcja ze środowiskiem na podstawie zebranych informacji. W przeciwieństwie do wymienionych wcześniej rodzajów, w uczeniu przez wzmacnianie nie przygotowuje się zestawu danych uczących, tylko środowisko, z którego model będzie zbierał dane automatycznie. Jego celem jest zmaksymalizowanie zwracanej przez nie nagrody (model jest nagradzany). Środowisko może być zależne od celu nauki. W przypadku uczenia programu grającego w gry będzie to gra, wraz z jej zasadami, lub prawdziwy świat, w przypadku programu uczącego się sterować łazikiem.

W uczeniu przez wzmacnianie wyróżnia się 3 główne elementy:

- Środowisko – to zadanie lub symulacja, z którym algorytm (w tym kontekście nazywany również agentem albo graczem) wchodzi w interakcję. Celem uczenia przez wzmacnianie jest maksymalizacja nagrody zwracanej przez środowisko, czyli nauczanie agenta osiągania w nim najwyższego wyniku, np. wygrania największej ilości gier albo osiągnięcia największej stopy zwrotu (jeśli środowisko jest symulacją giełdy).
- Agent – to element, który wchodzi w interakcję ze środowiskiem. Zadaniem agenta jest maksymalizowanie nagrody, czyli nauczanie się najkorzystniejszego oddziaływania ze środowiskiem. Za zachowanie agenta odpowiada tzw. polityka, czyli funkcja, zwracająca akcję. Jako politykę najczęściej stosuje się sieć neuronową.
- Bufor – magazyn danych przechowujący informacje zebrane przez agenta w trakcie uczenia, które są następnie wykorzystane do jego wytrenowania.

Uczenie przez wzmacnianie można podzielić na elementy: system wejściowy i wyjściowy, nagroda, środowisko AI, proces decyzyjny Markowa, szkolenie i wnioskowanie .
przykład:

- Q-Learning

Uczenie głębokie (Deep Learning):

Uczenie głębokie, często oparte na głębokich sieciach neuronowych, to rodzaj uczenia maszynowego, który jest wysoce zaawansowany i wykorzystywany do rozwiązywania skomplikowanych problemów, takich jak rozpoznawanie obrazów, przetwarzanie języka naturalnego i wiele innych. Modele te składają się z wielu warstw neuronów, które są w

stanie automatycznie ekstrahować cechy z danych.

- Neural Networks (NN)
- Recurrent Neural Networks (RNN)

[28][70][71][72][73][74]

2.4.1. Uczenie nienadzorowane (Unsupervised Learning)

Klastrowanie Np. grupowanie/klastrowanie (clustering) jest jednym z metod sztucznej inteligencji. Grupowanie, znane również jako klasyfikacja lub analiza skupień, to technika przetwarzania danych wykorzystywana w dziedzinie uczenia maszynowego i sztucznej inteligencji. Polega na dzieleniu zbioru danych na grupy (skupienia) na podstawie podobieństwa między jego elementami. W analizie danych i uczeniu maszynowym termin "clustering" jest szeroko używany i odnosi się do procesu tworzenia grup (klas) obiektów na podstawie pewnych kryteriów, takich jak odległość między obiektami. Celem jest stworzenie skupisk obiektów, które są podobne do siebie wewnątrz klastra, ale różnią się od obiektów w innych klastrach. Istnieje cały szereg algorytmów zastosowań klastrowania w sztucznej inteligencji.

Klastrowanie to dla uczenia maszynowego nienadzorowanego. Możesz również usłyszeć, że nazywa się to analizą skupień ze względu na sposób działania tej metody. Użycie algorytmu grupowania oznacza, że dasz algorytmowi dużo danych wejściowych bez etykiet i pozwolisz mu znaleźć dowolne grupy w danych, jakie tylko może. Grupy te nazywane są klastrami. Klaster to grupa punktów danych, które są do siebie podobne na podstawie ich relacji do otaczających punktów danych. Kiedy zaczynasz od danych, o których nic nie wiesz, klastrowanie może być dobrym miejscem na uzyskanie wglądu. Istnieją różne typy algorytmów grupowania:

- Oparte na gęstości. W klastrowaniu opartym na gęstości dane są grupowane według obszarów o dużej koncentracji punktów danych otoczonych obszarami o niskiej koncentracji punktów danych. Zasadniczo algorytm znajduje miejsca gęste z punktami danych i wywołuje te klastry.
 - Oparta na dystrybucji. W przypadku podejścia klastrowego opartego na dystrybucji wszystkie punkty danych są uważane za części klastra na podstawie prawdopodobieństwa, że należą do danego klastra. Działa to w ten sposób: istnieje punkt środkowy i wraz ze wzrostem odległości punktu danych od środka maleje prawdopodobieństwo, że będzie on częścią tego klastra.
 - Oparty na centroidach. Jest trochę wrażliwy na parametry początkowe jakie mu podajesz, ale jest szybki i wydajny. Algorytmy tego typu oddzielają punkty danych na podstawie wielu centroidów w danych. Każdy punkt danych jest przypisany do klastra na podstawie kwadratu odległości od środka ciężkości. Jest to najczęściej stosowany rodzaj grupowania.
 - Oparte na hierarchii. Klastrowanie oparte na hierarchii jest zwykle używane w przypadku danych hierarchicznych, takich jak można uzyskać z firmowej bazy danych lub taksonomii. Buduje drzewo klastrow, dzięki czemu wszystko jest zorganizowane od góry do dołu.
- [75]

Najbardziej znane algorytmy sztucznej inteligencji bazujące na klastrowaniu to:

- **K-means:** K-means jest jednym z najbardziej rozpowszechnionych algorytmów klastrowania ze względu na swoją prostotę i wydajność. Jest często używany do analizy danych w różnych dziedzinach, takich jak marketing, przetwarzanie obrazów, przetwarzanie języka naturalnego i wiele innych. K oznacza klasteryzację jest powszechnym algorytmem wśród naukowców danych. Jest to rodzaj algorytmu opartego na centroidach z prostymi i bezpośrednimi właściwościami. Co więcej, jest to algorytm uczenia bez nadzoru. Dzięki temu algorytmowi można zminimalizować wariancję punktu danych w klastrze. Wiele osób, które rozpoczynają nienadzorowane uczenie maszynowe, zaczyna od algorytmów klasteryzacji K. Najlepsze wyniki znajdziesz z tymi algorytmami klasteryzacji, zawierającymi małe zestawy danych. To dlatego, że ten algorytm powtarza wszystkie punkty danych. To wskazuje, że jeśli masz ogromną ilość danych, trzeba będzie więcej czasu.

- **Hierarchiczne klastrowanie:** Hierarchiczne klastrowanie jest popularne ze względu na możliwość tworzenia hierarchicznej struktury klastrowania. Hierarchiczne grupy klastrowania są klastry w zależności od odległości od jednych danych do drugich. Klastry te mają różne typy: – **Aglomeracyjne** W tej metodzie klasteryzacji jeden punkt danych działający jako klaster przyciąga inne podobne punkty danych, które stają się klastrami. – **Podzielna** Z drugiej strony, metoda dywizjonistyczna traktuje wszystkie punkty danych jako jeden klaster, a następnie rozdziela każdy z nich tworząc nowe klastry. Metoda ta jest przeciwieństwem metody aglomeracyjnej i działa poprzez łączenie istniejących klastrow, tworzenie macierzy odległości i łączenie ich razem.

- **Density-Based Clustering, DBSCAN:** Algorytm DBSCAN jest stosowany w przypadkach, gdy dane mają złożone kształty klastrow i zawierają punkty odstające. Jest wykorzystywany w analizie gęstości danych. W tej metodzie, algorytmy klasteryzacji będzie wymagać gęstości danych do tworzenia klastrow reprezentujących przestrzeń danych. Gdy przestrzeń lub region rośnie gęstość, to region staje się klastrem. Regiony o mniejszej gęstości lub z minimalną ilością danych są traktowane jako wartości odstające lub szum. Jest to dobry algorytm do wyszukiwania konturów w zbiorze danych. Znajduje klastry o dowolnym kształcie w oparciu o gęstość punktów danych w różnych regionach. Oddziela regiony według obszarów o małej gęstości, dzięki czemu może wykryć wartości odstające pomiędzy klastrami o dużej gęstości. Algorytm ten jest lepszy niż k-średnie, jeśli chodzi o pracę z danymi o dziwnych kształtach. DBSCAN wykorzystuje dwa parametry do określenia sposobu definiowania klastrow: minPts (minimalna liczba punktów danych, które muszą być zgrupowane razem, aby obszar można było uznać o dużej gęstości) i eps (odległość używana do określenia, czy punkt danych znajduje się w tym samym obszarze, co inne punkty danych).

Wybór właściwych parametrów początkowych ma kluczowe znaczenie dla działania tego algorytmu.

- **Fuzzy C-means (FCM)** ,Klasteryzacja rozmyta: FCM jest stosowany tam, gdzie ist-

nieje potrzeba uwzględnienia stopnia przynależności punktów danych do klastrów. Jest używany w analizie obrazów, rozpoznawaniu wzorców i kontroli procesów. W tej metodzie, wyrównanie punktów danych nie jest decydujące. W klasteryzacji rozmytej, punkt danych może łączyć się z więcej niż jednym klastrem. Wynik klastra jest prawdopodobieństwem, że punkt danych grupuje się w ramach danej grupy. Mechanizm działania tej metody klasteryzacji jest podobny do klasteryzacji metodą K means. Jednak parametry, które wymagają obliczeń są inne.

- ISODATA, Algorytm ISODATA działa w oparciu o iteracyjny proces, w którym początkowo dzieli dane na początkowe klastry, a następnie łączy i dzieli klastry w trakcie kolejnych iteracji na podstawie określonych kryteriów, takich jak minimalna i maksymalna liczba klastrów, minimalna liczba punktów w klastrze itp. Algorytm ten próbuje zoptymalizować klastry w miarę postępu iteracji.

- Gaussian Mixture Model, o popularny algorytm klastrowania probabilistycznego, który zakłada, że dane pochodzą z mieszaniny wielu rozkładów normalnych (Gaussowskich). Algorytm GMM jest stosowany do modelowania danych, które wydają się być wygenerowane przez wiele różnych źródeł lub komponentów. Jednym z problemów związanych z k-średnimi jest to, że dane muszą mieć format kołowy. Sposób, w jaki k-średnie oblicza odległość między punktami danych, ma związek ze ścieżką kołową, więc dane inne niż kołowe nie są poprawnie grupowane. Jest to problem, który rozwiązują modele mieszanin Gaussa. Nie potrzebujesz okrągłych danych, aby działało dobrze. Model mieszaniny Gaussa wykorzystuje wiele rozkładów Gaussa w celu dopasowania danych o dowolnym kształcie. Istnieje kilka pojedynczych modeli Gaussa, które działają jak warstwy ukryte w tym modelu hybrydowym. Zatem model oblicza prawdopodobieństwo, że punkt danych należy do określonego rozkładu Gaussa i do tego klastra będzie się zaliczał. Rozkład normalny, rozkład Gaussa to jeden z najważniejszych rozkładów prawdopodobieństwa, odgrywający ważną rolę w statystyce. W wielkim skrócie opisuje on sytuację w świecie, gdzie większość przypadków jest bliska średniemu wynikowi, a im dany wynik bardziej odchyła się od średniej tym jest mniej reprezentowany. Rozkład Gaussa jest opisany przez dwie główne charakterystyki: średnią (μ) i odchylenie standardowe (σ). Średnia określa, gdzie znajduje się wartość oczekiwana rozkładu, natomiast odchylenie standardowe mierzy, jak "rozciągnięty" lub "skoncentrowany" jest rozkład wokół średniej.

- BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) to algorytm klastrowania hierarchicznego, który jest używany do analizy dużych zbiorów danych. BIRCH jest popularny w analizie danych, zwłaszcza w sytuacjach, gdy musisz efektywnie klastrować duże ilości danych. Dzieli dane na małe podsumowania, które są grupowane zamiast oryginalnych punktów danych. Podsumowania zawierają możliwie najwięcej informacji o rozkładzie punktów danych. Algorytm ten jest powszechnie używany z innymi

algorytmami grupowania, ponieważ w podsumowaniach generowanych przez BIRCH można zastosować inne techniki grupowania. Główną wadą algorytmu BIRCH jest to, że działa on tylko na wartościach danych liczbowych. Nie możesz tego użyć do wartości kategorycznych, chyba że wykonasz pewne przekształcenia danych.

Zalety:

- Skalowalność: Jest w stanie obsłużyć duże zbiory danych, co jest istotne w przypadku przetwarzania big data.
- Minimalna złożoność pamięciowa: BIRCH przechowuje skondensowane reprezentacje danych, co ogranicza zużycie pamięci.
- Hierarchiczne klastrowanie: Tworzy hierarchię klastrów, co pozwala na analizę danych na różnych poziomach granularności.

- Mean Shift: Ten algorytm jest używany do automatycznego znajdowania centrów klastrów na podstawie gęstości danych. Jest stosowany w rozpoznawaniu obrazów i śledzeniu obiektów. Algorytm Mean Shift jest używany do klastrowania i wykrywania lokalnych maksimum w funkcjach gęstości danych. Jest to algorytm klastrowania oparty na przesunięciu średniej wartości. W algorytmie Mean Shift, początkowo losowo wybierany jest zestaw punktów danych nazywanych "środkami klastrów". Następnie każdy punkt danych jest przesuwany w kierunku średniej ważonej innych punktów danych, gdzie waga każdego punktu zależy od jego odległości od punktu, który przemieszcza się. Proces ten jest powtarzany do momentu, gdy punkty danych nie przesuwają się znacząco. To kolejny algorytm, który jest szczególnie przydatny do obsługi obrazów i komputerowego przetwarzania obrazu. Przesunięcie średnie jest podobne do algorytmu BIRCH, ponieważ znajduje również skupienia bez ustawienia początkowej liczby skupień. Jest to algorytm grupowania hierarchicznego, ale jego wadą jest to, że nie skaluje się dobrze podczas pracy z dużymi zbiorami danych. Działa poprzez iterację po wszystkich punktach danych i przesuwanie ich w stronę trybu. Trybem w tym kontekście jest obszar o dużej gęstości punktów danych w regionie. Dlatego możesz usłyszeć, że ten algorytm jest nazywany algorytmem wyszukiwania trybu. Przejdzie przez ten iteracyjny proces z każdym punktem danych i przesunie go bliżej miejsc, gdzie znajdują się inne punkty danych, dopóki wszystkie punkty danych nie zostaną przypisane do klastra.

- Agglomerative Clustering: Ten algorytm jest często używany do tworzenia hierarchii klastrów w danych, co pozwala na analizę danych na różnych poziomach granularności.

Działanie algorytmu Agglomerative Hierarchical Clustering polega na początkowym traktowaniu każdego punktu danych jako osobnego klastra. Jest to forma grupowania oddolnego, w której każdy punkt danych jest przypisany do własnego klastra. Następnie te klastry łączą się ze sobą. W każdej iteracji podobne klastry są łączone, aż wszystkie punkty danych staną się częścią jednego dużego klastra głównego. Iteracyjnie łączy się najbliższe

klastry w jeden, aż wszystkie punkty danych zostaną połączone w jeden duży klaster lub zostanie osiągnięta wcześniej zdefiniowana liczba klastrów.

W wyniku działania algorytmu Agglomerative Hierarchy Clustering tworzona jest hierarchia klastrów, która może być reprezentowana w formie drzewa

Algorytm ten ma wiele zastosowań, w tym w analizie obrazów, analizie tekstu, biologii, analizie społeczności w mediach społecznościowych i wielu innych dziedzinach, gdzie istnieje potrzeba hierarchicznego modelowania struktury danych. Jeden z głównych atutów algorytmu Agglomerative Hierarchy Clustering to możliwość analizy danych na różnych poziomach szczegółowości i identyfikowania klastrów o różnych rozmiarach i kształtach.

- **Spectral Clustering:** Spectral Clustering jest popularny w analizie obrazów, analizie tekstu i innych dziedzinach, gdzie występują nieliniowe zależności między danymi.

- **Algorytm OPTICS** (Ordering Points To Identify the Clustering Structure) to technika klastrowania i identyfikacji struktury klastrów w danych. Jest to rozszerzenie algorytmu DBSCAN (Density-Based Spatial Clustering of Applications with Noise). OPTICS jest używany do analizy danych przestrzennych, gdzie rozkład danych może być gęstościowo zmieniający się. Algorytm OPTICS nie tylko wykrywa klastry, ale także hierarchicznie uporządkowuje punkty danych wzdłuż osi gęstości, tworząc tzw. "diagram uporządkowania" (reachability plot). W tym diagramie punkty danych są rozmieszczone wzdłuż osi x w zależności od swojego stopnia dostępności (reachability) od innych punktów danych. Dzięki temu diagramowi można analizować strukturę klastrów oraz określać poziomy gęstości danych.

Algorytm OPTICS jest przydatny w przypadkach, gdy dane są zróżnicowane pod względem gęstości lub gdy klastry są nieregularne i różnią się wielkością. Dokonuje tego poprzez uporządkowanie punktów danych w taki sposób, aby najbliższe punkty były sąsiadami w kolejności. Ułatwia to wykrywanie klastrów o różnej gęstości. Algorytm OPTICS przetwarza każdy punkt danych tylko raz, podobnie jak DBSCAN (choć działa wolniej niż DBSCAN). Dla każdego punktu danych przechowywana jest także specjalna odległość, która wskazuje, że punkt należy do określonego klastra. Pomaga on również w identyfikacji punktów odstających oraz w hierarchicznym modelowaniu struktury danych.

Warto zauważyć, że algorytm OPTICS jest szczególnie użyteczny w analizie danych przestrzennych, takich jak dane geograficzne, ale może być stosowany również w innych dziedzinach, gdzie występują dane przestrzenne lub o zróżnicowanej gęstości.

- **Affinity Propagation:** Algorytm ten jest używany w zadaniach grupowania obrazów medycznych, analizie społeczności w mediach społecznościowych i wielu innych dziedzinach. Algorytm Affinity Propagation to technika klastrowania, która jest używana do identyfikowania grup w danych na podstawie podobieństwa między punktami danych. W algorytmie Affinity Propagation, każdy punkt danych komunikuje się z innymi punktami, przesyłając informacje o swoim podobieństwie do innych punktów. Na podstawie tych

komunikatów algorytm próbuje wyodrębnić klastry poprzez wybór punktów eksemplarzy, które najlepiej reprezentują klastry. Nie musisz mówić temu algorytmowi, ile klastrow ma się spodziewać w parametrach inicjujących. Gdy komunikaty są przesyłane pomiędzy punktami danych, odnajdywane są zestawy danych zwane wzorami, które reprezentują klastry. Przykład można znaleźć po tym, jak punkty danych przesłały sobie nawzajem komunikaty i osiągnęły konsensus co do tego, który punkt danych najlepiej reprezentuje klaster. Jeśli brak pewności, ilu klastrow się spodziewać, jak w przypadku problemu z widzeniem komputerowym, jest to świetny algorytm na początek. Jest to algorytm oparty na przesyłaniu wiadomości i obliczeniach iteracyjnych.

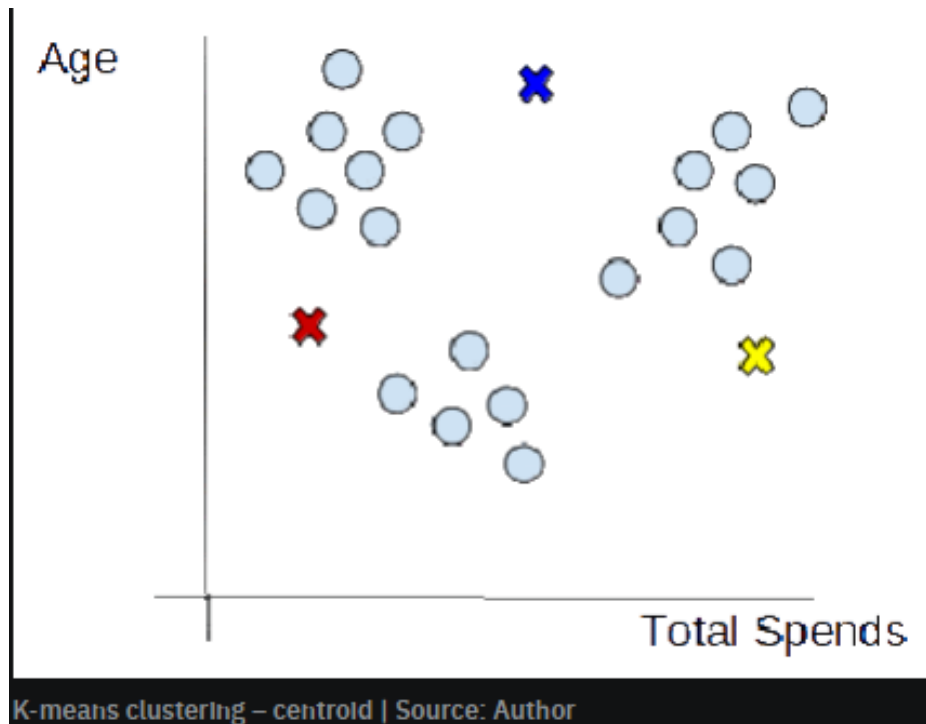
[76][77][75][78][79]

Algorytm grupowania K-średnich(K-means) .Grupowanie K-średnich jest najczęściej używanym algorytmem grupowania. Jest to algorytm oparty na centroidach i najprostszy algorytm uczenia się bez nadzoru. Algorytm ten próbuje zminimalizować wariancję punktów danych w klastrze. Średnie K najlepiej stosować w przypadku mniejszych zestawów danych, ponieważ iterują po wszystkich punktach danych. Oznacza to, że klasyfikacja punktów danych zajmie więcej czasu, jeśli w zbiorze danych znajduje się ich duża liczba.[75]

Jest to iteracyjny proces przypisywania każdego punktu danych do grup, w wyniku którego punkty danych są powoli grupowane w oparciu o podobne cechy. Celem jest zminimalizowanie sumy odległości między punktami danych a środkiem ciężkości klastra, aby zidentyfikować właściwą grupę, do której powinien należeć każdy punkt danych. Tutaj dzielimy przestrzeń danych na K klastrow i przypisujemy każdemu z nich średnią wartość. Punkty danych są umieszczane w skupieniach najbliższych średniej wartości tego skupienia. Dostępnych jest kilka wskaźników odległości, które można wykorzystać do obliczenia odległości. Jak działają K-średnie? Weźmy przykład, aby zrozumieć, jak krok po kroku działają K-średnie. Algorytm można podzielić na 4-5 kroków.

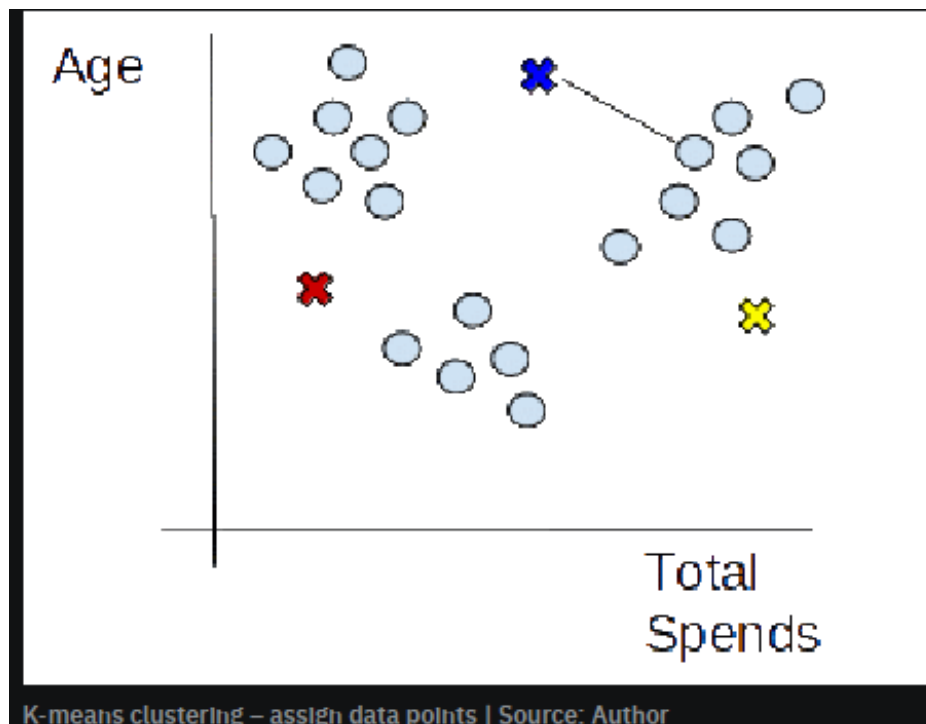
A. Wybór liczby klastrow Pierwszym krokiem jest zdefiniowanie liczby K skupień, w których będziemy grupować dane. Wybierzmy $K=3$.

B. Inicjowanie centroidów Centroid jest środkiem klastra, ale początkowo dokładny środek punktów danych będzie nieznany, dlatego wybieramy losowe punkty danych i definiujemy je jako centroidy dla każdego klastra. Zainicjujemy 3 centroidy w zbiorze danych.



Rysunek 2.3. k1 Grupowanie K-średnich[75]

C.Przypisz punkty danych do najbliższego klastra Teraz, gdy centroidy zostały zainicjowane, następnym krokiem jest przypisanie punktów danych X_n do ich najbliższego środka ciężkości klastra C_k



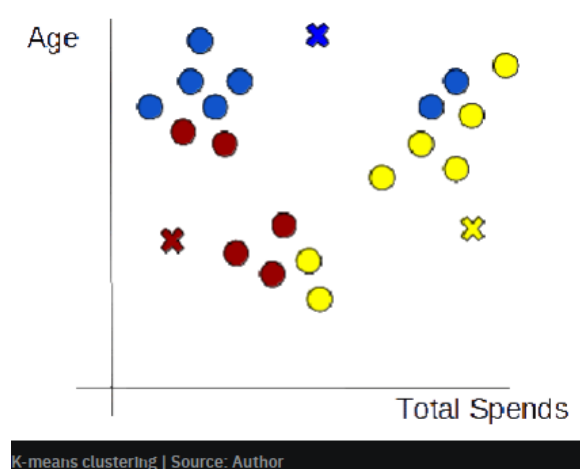
Rysunek 2.4. k2 Grupowanie K-średnich przypisanie danych[75]

W tym kroku najpierw obliczymy odległość między punktem danych X a środkiem ciężkości C , korzystając z metryki odległości euklidesowej. Metryka odległości euklidesowej:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Metryka odległości euklidesowej. Wzór [75]

Następnie wybierz klastery dla punktów danych, w których odległość między punktem danych a środkiem ciężkości jest minimalna.

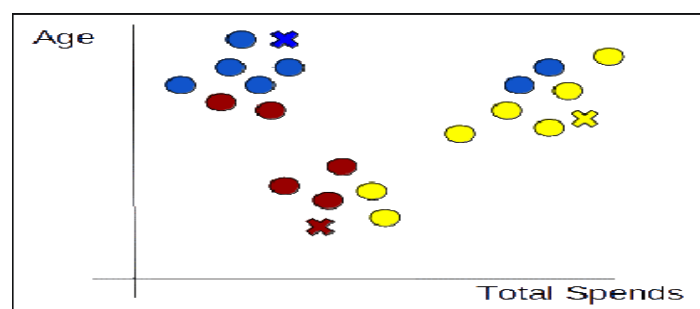


Rysunek 2.5. k4 Grupowanie K-średnich, dane przypisane do klastrów [75]

D. Zainicjuj ponownie centroidy Następnie ponownie zainicjujemy centroidy, obliczając średnią wszystkich punktów danych tego klastra.

$$C_i = \frac{1}{|N_i|} \sum X_i$$

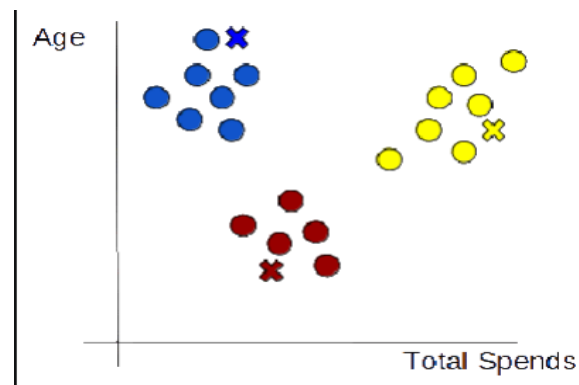
Obliczamy centroidy. Wzór [75]



Rysunek 2.6. k6 Grupowanie K-średnich, dane przypisane do klastrów [75]

E.Powtórz kroki 3 i 4

Będziemy powtarzać kroki 3 i 4, aż uzyskamy optymalne centroidy, a przypisanie punktów danych do odpowiednich klastrów nie będzie się już zmieniać.

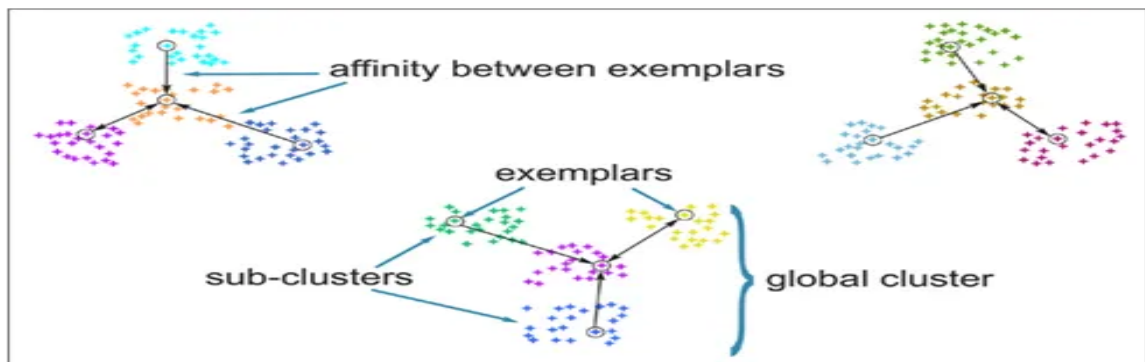


Rysunek 2.7. k7 Grupowanie K-średnich,dane przypisane do klastrów[75]

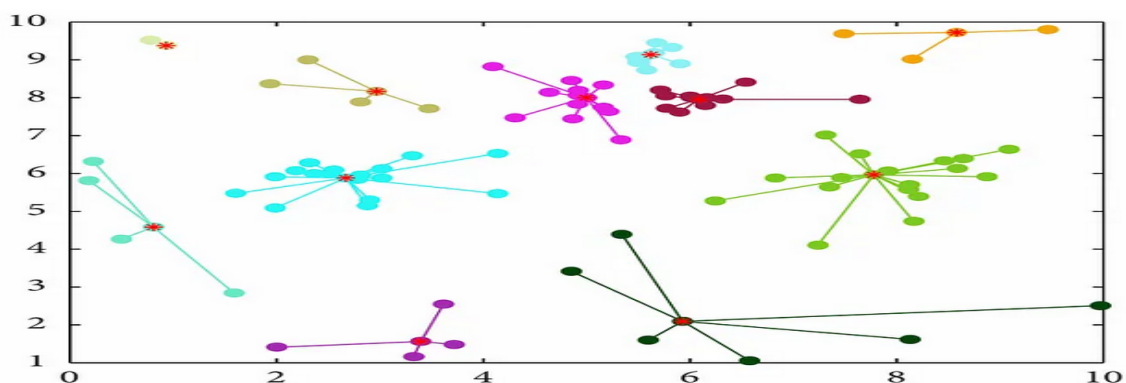
Does this iterative process sound familiar? Well, K-means follows the same approach as Expectation-Maximization(EM). [75]

affinity propagation (AP) affinity propagation to algorytm grupowania oparty na koncepcji „przekazywania wiadomości” pomiędzy punktami danych. W przeciwieństwie do algorytmów grupowania, takich jak k-średnie, affinity propagation nie wymaga określenia lub oszacowania liczby klastrów przed uruchomieniem algorytmu.

Przykłady to konkretne punkty danych, które służą jako reprezentacje lub prototypy dla każdego klastra. Zasadniczo przykład to punkt danych w klastrze, który najlepiej podsumowuje lub charakteryzuje inne punkty danych w tym klastrze.



Rysunek 2.8. b1 Grupowanie affinity propagation, wskazanie przykładów (exemplars) [80]



Rysunek 2.9. b2 Grupowanie affinity propagation, wskanie klastrów [80]

Każda linia w wizualizacji grupowania affinity propagation może łączyć punkt danych z odpowiadającym mu przykładem, demonstrując związek między członkami klastra a ich reprezentatywnym przykładem. W przeciwieństwie do centroidów w innych metodach grupowania, przykłady są rzeczywistymi punktami danych w zbiorze danych. K-średnie, chociaż czasami określane jako metoda grupowania oparta na przykładach, zasadniczo różnią się pod tym względem od affinity propagation. W przypadku K-średnich centroidy reprezentujące klastry są obliczane na podstawie średniej punktów danych w każdym klastrze i mogą nie pokrywać się z rzeczywistymi punktami danych. To rozróżnienie podkreśla jedną z kluczowych różnic między K-średnimi a affinity propagation: podczas gdy propagacja powinowactwa wybiera przykłady z istniejących punktów danych, K-średnie

2. Przegląd

oblicza centroidy, które mogą nie odpowiadać żadnemu konkretnemu punktowi danych, co skutkuje innym podejściem do charakteryzowania i tworząc skupiska.

Opis koncepcji/teorii:

A. Dobrze skaluje, ponieważ wykorzystuje macierze

B. Do grupowania wykorzystuje 4 macierze, tj. macierz podobieństwa, macierz dostępności, macierz odpowiedzialności, macierz kryteriów.

C. Zestawy „Wzorów”: obiekty, które kończą się tym samym egzemplarzem, są grupowane razem.

Cechy AP:

- Nie nakłada żadnych ograniczeń na macierz podobieństwa, podobieństwa mogą być nawet asymetryczne.

Wszystkie punkty jednocześnie są rozważane jako potencjalne archetypy.

Liczba skupień nie jest zadawana wprost, wynika z wartości „preferencji” przypisanej każdemu punktowi (preferencja określa jak silnie dany punkt powinien być preferowany jako archetyp w ostatecznym wyniku).

Przykład obliczeń:

Niech każdy obiekt będzie reprezentowany jako punkt w pięciowymiarowej przestrzeni rzeczywistej. Warto zauważyć, że w tym przykładzie zmienne są w tej samej skali. Na ogół jednak zmienne znajdują się w różnych skalach i należy je znormalizować.

Participant	Tax Rate	Fee	Interest Rate	Quantity Limit	Price Limit
Alice	3	4	3	2	1
Bob	4	3	5	1	1
Cary	3	5	3	3	3
Doug	2	1	3	3	2
Edna	1	1	3	2	3

Rysunek 2.10. b3 Tabela z danymi wejściowymi)[81]

W pierwszym kroku należy wyznaczyć macierz podobieństwa (SSS) – ang. similarity matrix. Z wyjątkiem elementów na przekątnej, każda komórka w tej macierzy jest obliczana jako ujemna odległość euklidesowa podniesiona do kwadratu, czyli

$$s_{ik} = -\|\mathbf{x}_i - \mathbf{x}_k\| \quad (1)$$

Np. dla podobieństwa między Alice i Bobem, suma kwadratów różnic wynosi:

$$(3-4)^2 + (4-3)^2 + (3-5)^2 + (2-1)^2 + (1-1)^2 = 7 \quad (2)$$

Zatem wartość podobieństwa wynosi $s_{12}=s_{21}=-7$

Przekątna macierzy SSS (tj. s_{ii}) jest szczególnie ważna, ponieważ reprezentuje preferencje obiektu, czyli to, jak prawdopodobne jest, że dany obiekt stanie się archetypem. Algorytm będzie zbiegał do małej liczby skupień, jeśli dla przekątnej zostanie wybrana mniejsza wartość i odwrotnie. Dlatego w naszym przykładzie elementy diagonalne macierzy podobieństwa wypełniamy liczbą -22, czyli najniższą liczbą spośród wyliczonych podobieństw.

Macierz podobieństwa (S)

Participant	Alice	Bob	Cary	Doug	Edna
Alice	-22	-7	-6	-12	-17
Bob	-7	-22	-17	-17	-22
Cary	-6	-17	-22	-18	-21
Doug	-12	-17	-18	-22	-3
Edna	-17	-22	-21	-3	-22

Rysunek 2.11. b4 macierz podobieństw)[81]

Następnie wykonywane są dwa kroki przekazywania wiadomości, które aktualizują dwie macierze: Macierz odpowiedzialności (R) – ang. responsibility matrix, która określa jak dobrze x_k jest przystosowany do pełnienia roli wzorca dla x_i , w stosunku do innych kandydatów na wzorce dla x_i .

Macierz dostępności (AAA) – ang. availability matrix, która określa jak „odpowiednie” byłoby dla x_i wybranie x_k jako swojego wzorca, biorąc pod uwagę preferencje innych punktów dla x_k jako wzorca

dla macierzy odpowiedzialności (R) wzór:

$$R_{ik} = S_{ik} - \max(A_{ik} - S_{ik}) \quad (3)$$

Np. dla Boba (kolumna) i Alice (wiersz) wynosi -1, co jest różnicą ich podobieństwa -7 i maksimum pozostałych podobieństw dla wiersza Alice -6, czyli $R_{12} = -7 - (-6)$

Np. samodostępność Alice jest sumą dodatnich odpowiedzialności kolumny Alice z wyłączeniem samoodpowiedzialności Alice czyli $a_{11} = 10 + 11 + 0 + 0 = 21$. Podobnie dostępność Boba (kolumna) dla Alice (wiersz) to odpowiedzialność własna Boba plus suma pozostałych pozytywnych odpowiedzialności Boba w kolumnie z wyłączeniem odpowiedzialności Boba wobec Alice, czyli $a_{12} = -15 + 0 + 0 + 0 = -15$

Macierz odpowiedzialności (R)

Participant	Alice	Bob	Cary	Doug	Edna
Alice	-16	-1	1	-6	-11
Bob	10	-15	-10	-10	-15
Cary	11	-11	-16	-12	-15
Doug	-9	-14	-15	-19	9
Edna	-14	-19	-18	14	-19

Rysunek 2.12. b5 macierz odpowiedzialności R)[81]

Macierz dostępności (A)

Participant	Alice	Bob	Cary	Doug	Edna
Alice	21	-15	-16	-5	-10
Bob	-5	0	-15	-5	-10
Cary	-6	-15	1	-5	-10
Doug	0	-15	-15	14	-19
Edna	0	-15	-15	-19	9

Rysunek 2.13. b6 macierz dostępności A)[81]

Macierz kryterium (C)

Participant	Alice	Bob	Cary	Doug	Edna
Alice	5	-16	-15	-11	-21
Bob	5	-15	-25	-15	-25
Cary	5	-26	-15	-17	-25
Doug	-9	-29	-30	-5	-10
Edna	-14	-34	-33	-5	-10

Rysunek 2.14. b7 macierz kryterium C)[81]

Np. wartość kryterialna Boba (kolumna) dla Alice (wiersz) jest sumą odpowiedzialności i dostępności Boba dla Alicji, czyli $c_{12} = -1 + (-15) = -16$. Najwyższa wartość kryterium w każdym wierszu jest oznaczana jako archetyp. Wiersze, które mają ten sam archetyp, znajdują się w tym samym skupieniu. Zatem w naszym przykładzie Alice, Bob i Cary tworzą jedno skupienie, podczas gdy Doug i Edna tworzą drugie

[81][82][80][83]

kolejny algorytm klastrowania to:

Agglomerative Clustering (Klastrowanie aglomeracyjne) Klastrowanie aglomeracyjne to algorytm uczenia maszynowego bez nadzoru, który grupuje dane w klastry. Jest to podejście oddolne, oparte na hierarchicznym grupowaniu, w którym każdy punkt danych jest przypisywany do własnego klastra, a następnie klastry są łączone w miarę postępu algorytmu. Każdy klaster w skupieniu aglomeracyjnym jest tworzony na podstawie odległości między punktami danych, którą można obliczyć przy użyciu różnych technik, takich jak odległość euklidesowa. Algorytm rozpoczyna się od przypisania każdego punktu danych do własnego klastra. Następnie algorytm wyszukuje dwa najbliższe skupienia i łączy je w jeden klaster. Proces ten powtarza się do momentu zgrupowania wszystkich punktów w jedno skupienie lub do osiągnięcia pożądanej liczby skupień.

Poniżej przedstawiono etapy grupowania aglomeracyjnego.

- Zaczynamy od przypisania każdego punktu danych do własnego klastra.
- Następnie obliczamy odległość pomiędzy każdą parą skupień i wybieramy parę skupień o najmniejszej odległości.
- Następnie łączymy parę skupień o najmniejszej odległości w jedno skupienie i aktualizujemy odległość pomiędzy nowo powstałym skupieniem a każdym innym skupieniem.
- Powtarzamy kroki 2 i 3, aż wszystkie punkty danych znajdą się w jednym klastrze.

Aby rozwiązać numeryczny przykład grupowania aglomeracyjnego, weźmy punkty A (1, 1), B (2, 3), C (3, 5), D (4,5), E (6,6) i F(7,5) i je pogrupuje.

najpierw utworze macierz odległości składającą się z odległości między każdym punktem w zbiorze danych. Macierz odległości wygląda następująco.

	A	B	C	D	E	F
A	0	2.236068	4.472136	5	7.071068	7.211103
B	2.236068	0	2.236068	2.828427	5	5.385165
C	4.472136	2.236068	0	1	3.162278	4
D	5	2.828427	1	0	2.236068	3
E	7.071068	5	3.162278	2.236068	0	1.414214
F	7.211103	5.385165	4	3	1.414214	0

Rysunek 2.15. b8 macierz odległości)[84]

. Do stworzenia powyższej tabeli wykorzystaliśmy odległość euklidesową. Po obliczeniu macierzy odległości zaczniemy grupować punkty danych, stosując podejście pojedynczego połączenia Krok 1: Najpierw rozważymy każdy punkt danych jako pojedynczy klaster. Następnie zaczniemy łączyć klastry. Krok 2: Aby połączyć poszczególne klastry, możemy rozważyć następujące punkty.

2. Przegląd

Punkt A jest najbliższym punktu B.

Punkt B znajduje się w podobnej odległości od punktów A i C.

Punkt C jest najbliższym punktu D.

Punkt D jest najbliższym punktu C.

Punkt E jest najbliższym punktu F.

Punkt F jest najbliższym punktu E.

Z powyższych punktów możemy połączyć (C, D) i (E, F) jako klastry. Nazwiemy je odpowiednio CD i EF. Ponieważ mamy niejednoznaczność punktów A i B, nie łączymy ich i traktujemy jako osobne skupienia

Krok 3: Teraz obliczymy minimalną odległość pomiędzy klastrami A, B, CD i EF. Możesz to zaobserwować

Gromada A jest najbliższym B.

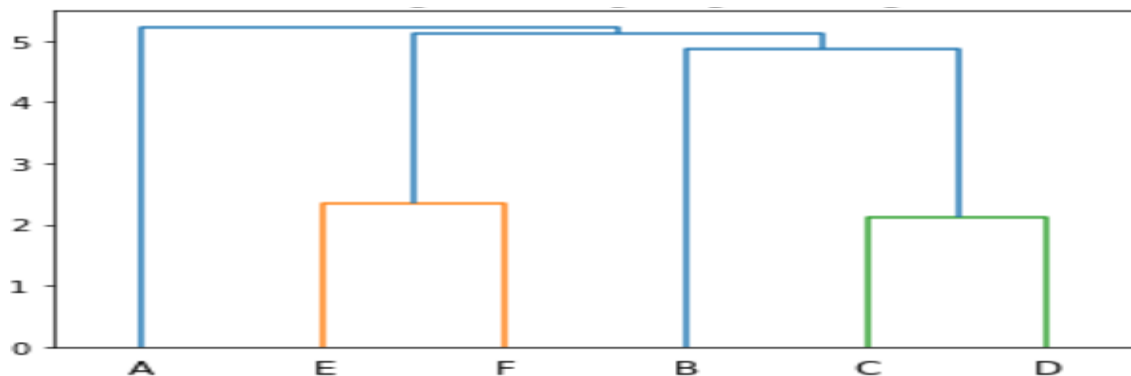
Gromada B jest najbliższym A i CD.

Klaster EF jest najbliższym CD.

Korzystając z powyższych informacji, połączmy B i CD i nazwijmy klaster BCD.

Krok 4: Następnie klaster BCD znajduje się w tej samej odległości od A i EF. Dlatego najpierw połączymy BCD i EF, tworząc BCDEF

Krok 5: Na koniec połączymy A i BCDEF, tworząc klaster ABCDEF. Wykonując powyższe kroki, otrzymamy następujący dendrogram.



Rysunek 2.16. b9 dendrogram)[84]

.

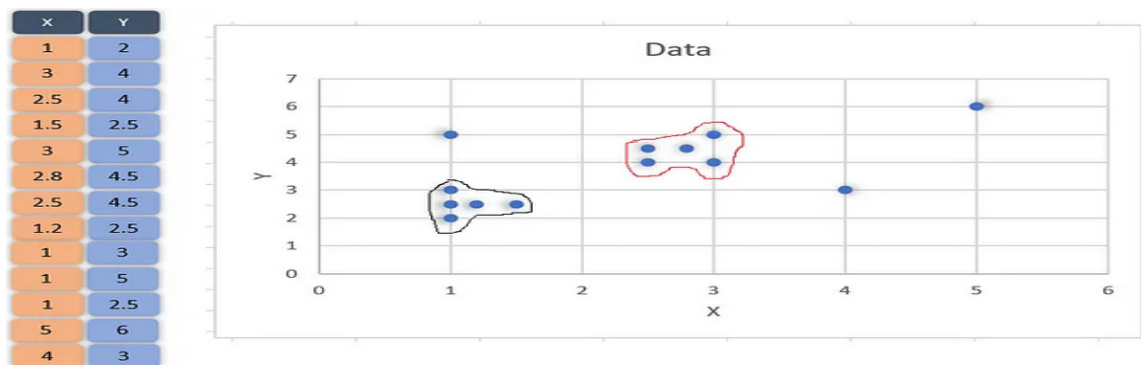
[84][85]

kolejny algorytm klastrowania to:

DBSCAN Algorytm DBSCAN (ang. Density Based Spatial Clustering of Application with Noise) jest algorytmem zaliczanym do klasy algorytmów gęstościowych. DBSCAN jest jednym z najpopularniejszych oraz najczęściej cytowanych algorytmów grupowania. DBSCAN

oznacza aplikację klastrowania przestrzennego opartą na gęstości z szumem. Jest to algorytm nienadzorowanego uczenia maszynowego, który tworzy klastry na podstawie gęstości punktów danych lub odległości, w jakiej znajdują się dane. To powiedziawszy, punkty znajdujące się poza gęstymi regionami są wykluczane i traktowane jako szum lub wartości odstające. Ta cecha algorytmu DBSCAN sprawia, że idealnie nadaje się on do wykrywania wartości odstających i tworzenia skupień o dowolnym kształcie.

Weźmy zbiór danych składający się z 13 punktów, jak pokazano i wykreślono poniżej:



Rysunek 2.17. b10 dane wejściowe)[86]

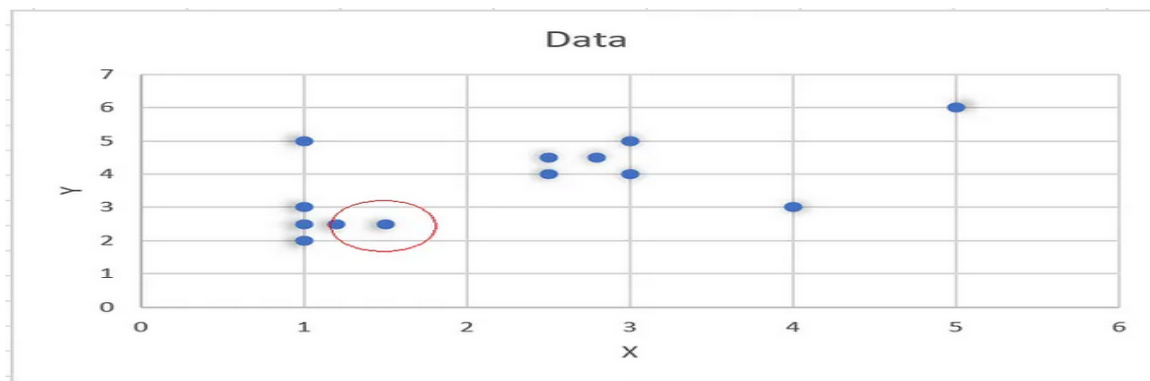
Dane dwuwymiarowe są prezentowane w celu łatwej wizualizacji i zrozumienia, w przeciwnym razie DBSCAN może również obsługiwać dane wielowymiarowe. Możliwe skupienia z danych zostały zaznaczone na powyższym wykresie, aby zwizualizować skupienia, które chcemy. Punkty (1,5) (4,3) (5,6) na powyższym wykresie wykraczają poza oznaczenia i dlatego należy je traktować jako wartości odstające. Algorytm DBSCAN powinien w rzeczywistości tworzyć skupienia i wykluczać wartości odstające, tak jak to zrobiliśmy na wykresie.

DBSCAN jako wejście przyjmuje dwa parametry. Pierwszy z nich to minimalna ilość punktów wymagana do utworzenia grupy oznaczany jako n lub minPts , natomiast drugi to maksymalny promień sąsiedztwa Eps . Samo sąsiedztwo (lub otoczenie) jest to zbiór punktów D leżących w odległości mniejszej bądź równej Eps od danego punktu p .

Rozważmy na przykład punkt (1.5,2.5). Jeśli przyjmiemy $\text{eps} = 0.3$, wówczas okrąg wokół punktu o promieniu $= 0.3$ będzie zawierał w sobie tylko jeden inny punkt (1.2,2.5), jak pokazano poniżej:

Dla (1,5, 2,5), gdy $\text{eps} = 0,3$, liczba punktów sąsiedzkich wynosi tylko jeden. W DBSCAN każdy punkt jest sprawdzany pod kątem tych dwóch parametrów, a decyzja o grupowaniu jest podejmowana w sposób opisany w poniższych krokach:

- Wybierz wartość dla eps i MinPts
- Dla konkretnego punktu danych (x) oblicz jego odległość od każdego innego punktu



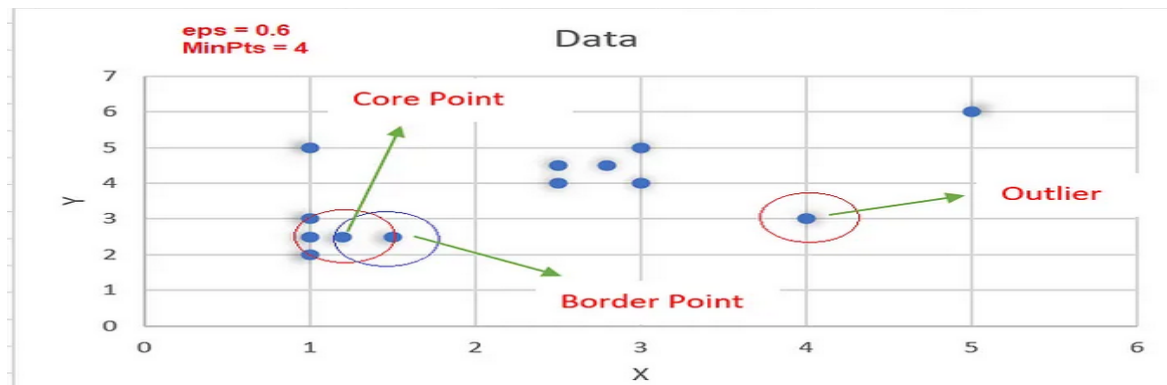
Rysunek 2.18. b11 eps 0.3 dla punktu (1.5,2.5) [86]

danych

- Znajdź wszystkie punkty sąsiedztwa x , które mieszczą się w okręgu o promieniu (eps) lub po prostu których odległość od x jest mniejsza lub równa eps.
- Traktuj x jako odwiedzone i jeśli liczba punktów sąsiedzkich wokół x jest większa lub równa MinPts, wówczas traktuj x jako punkt centralny i jeśli nie jest on przypisany do żadnego klastra, utwórz nowy klaster i przypisz go do niego.
- Jeżeli liczba punktów sąsiedzkich wokół x jest mniejsza niż MinPts i ma on punkt centralny w swoim sąsiedztwie, należy go potraktować jako punkt graniczny.
- Uwzględnij wszystkie punkty połączone z gęstością jako pojedynczy klaster.
- Powtórz powyższe kroki dla każdego nieodwiedzonego punktu w zestawie danych i znajdź wszystkie punkty podstawowe, graniczne i odstające.

Jeśli liczba punktów sąsiedztwa wokół x jest większa lub równa MinPts, wówczas x jest traktowane jako punkt centralny, jeśli punkty sąsiedzkie wokół x są mniejsze niż MinPts, ale znajdują się blisko punktu rdzeniowego, wówczas x jest traktowane jako punkt graniczny. Jeśli x nie jest ani rdzeniem, ani punktem granicznym, wówczas x jest traktowane jako wartość odstająca. Poniższy wykres daje wyobrażenie na ten temat. Wybieramy $\text{eps} = 0,6$ i $\text{MinPts} = 4$, punkt oznaczony jako punkt główny ma 4 inne punkty ($\geq \text{MinPts}$) w swoim sąsiedztwie, a ten oznaczony jako punkt graniczny znajduje się w sąsiedztwie punktu rdzeniowego, ale ma tylko jeden punkt w swoim sąsiedztwie ($< \text{MinPts}$). Punkt odstający to taki, który nie jest ani punktem granicznym, ani punktem rdzenia.

[87][86][88]



Rysunek 2.19. b12 punkty graniczne, centralne i odstające dla $\text{eps}=6$ i $\text{minPts}=4$ [86]

kolejny algorytm klastrowania to:

Gaussian Mixture Model (GMM, Model mieszaniny Gaussa) Modele probabilistyczne wykorzystują modele mieszaniny Gaussa do szacowania gęstości i danych skupień. Model mieszaniny Gaussa łączy wiele rozkładów Gaussa. Innymi słowy, parametry rozkładu Gaussa obliczają średnią, wariancję i wagę każdego klastra. Stąd możemy określić prawdopodobieństwa każdego punktu należącego do każdego skupienia po poznaniu parametrów każdego punktu. W porównaniu z innymi technikami grupowania, klastrowanie GMM zapewnia pseudoprawdopodobieństwa przynależności do klastra, w przeciwieństwie do zwykłego przypisywania każdego elementu danych do pojedynczego identyfikatora klastra. Na przykład, jeśli istnieje $k = 3$ skupień, określony element danych może mieć pseudoprawdopodobieństwa skupień $[0,15, 0,10, 0,85]$, co wskazuje, że element najprawdopodobniej należy do klastra $\text{ID} = 2$. Jednak w porównaniu z innymi technikami GMM klastrowanie jest bardziej skomplikowane w implementacji i nieco bardziej skomplikowane w interpretacji. Podstawową ideą modelu jest to, że dane są modelowane za pomocą kilku mieszanin rozkładów Gaussa.

Jednowymiarowa funkcja gęstości prawdopodobieństwa rozkładu Gaussa jest następująca:

$f(x)$ - funkcja gęstości prawdopodobieństwa dla zmiennej losowej X w punkcie x .

μ - średnia (wartość oczekiwana) zmiennej losowej X .

σ - odchylenie standardowe zmiennej losowej X .

e - liczba Eulera (około 2,71828).

π - liczba pi (około 3,14159).

Wzór gęstości prawdopodobieństwa rozkładu normalnego:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

przykładowe obliczenia:

2. Przegląd

$\mu = 2.0$ (średnia) $\sigma = 1.0$ (odchylenie standardowe) $x = 3.0$ (punkt, w którym chcemy obliczyć gęstość prawdopodobieństwa)

Mamy wzór gęstości prawdopodobieństwa rozkładu normalnego:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Podstawiamy wartości:

$$\mu = 2.0, \sigma = 1.0, x = 3.0,$$

$$f(3.0) = \frac{1}{1.0 \cdot \sqrt{2\pi}} e^{-\frac{(3.0-2.0)^2}{2 \cdot 1.0^2}}$$

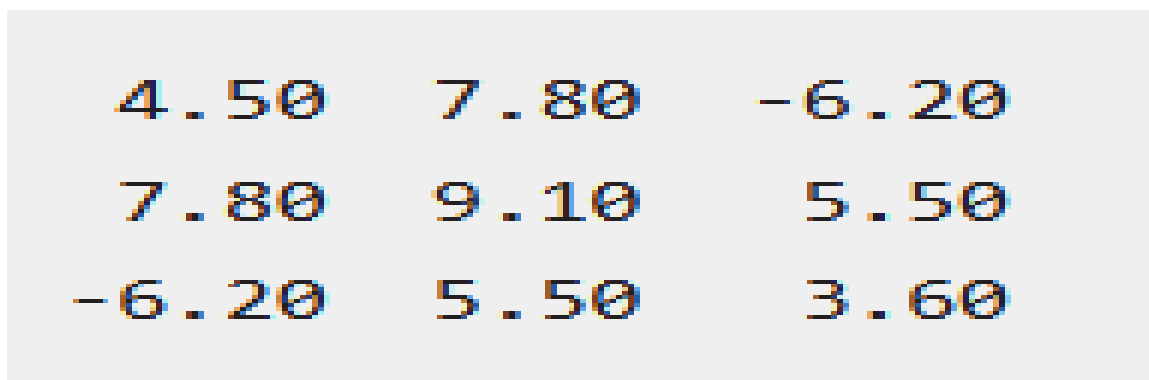
Teraz obliczamy to:

$$f(3.0) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}} \approx 0.24197072451914337$$

To jest wartość funkcji gęstości prawdopodobieństwa dla zmiennej losowej X w punkcie $x = 3.0$ dla rozkładu normalnego o średniej $\mu = 2.0$ i odchyleniu standardowym $\sigma = 1.0$. Możemy uznać tę wartość za prawdopodobieństwo, że zmienna X przyjmuje wartość bliską 3.0 w tym rozkładzie. można luźno interpretować jako prawdopodobieństwo odpowiedniej wartości x .

Wielowymiarowy rozkład Gaussa rozszerza tę koncepcję. Załóżmy, że zamiast patrzeć tylko na wzrost mężczyzn, patrzysz na wzrost i wagę każdego mężczyzny. Zatem element danych to wektor, który wygląda jak (69,50, 170,0), a nie tylko 69,50. Średnią danych wielowymiarowych będzie wektor o dwóch wartościach, np. (68,00, 175,25).

Aby opisać rozkład danych wielowymiarowych, zamiast pojedynczej wartości wariancji, będziesz mieć macierz kowariancji 2x2. Jeśli patrzysz na dane zawierające trzy wartości, np. dotyczące mężczyzn (wzrost, masa ciała, wiek), wówczas średnią danych będzie wektor z trzema wartościami, a macierz kowariancji będzie miała kształt 3x3. Załóżmy, że masz macierz kowariancji (wzrost, waga, wiek) w następujący sposób:



Rysunek 2.20. b13 macierz kowariancji)[89]

Trzy wartości na przekątnej to wariancje. Wartość 4,50 przy [0] [0] to wariancja samych wartości wzrostu. 9,10 w [1] [1] to wariancja samych wartości wag. 3,60 w [2] [2] to wariancja samych wartości wieku.

Wartość w [0] [1] i [1] [0] wynosi 7,80, co jest kowariancją zmiennej [0] (wzrost) i zmiennej [1] (waga). Kowariancja jest podobna do wariancji, ale kowariancja jest miarą zmienności dwóch zmiennych względem siebie. Wartość w [0] [2] i [2] [0] wynosi -6,20 i jest kowariancją zmiennej [0] (waga) i zmiennej [2] (wiek). Wartość w [1] [2] i [2] [1] jest kowariancją wartości wagi i wieku. Ponieważ macierz kowariancji przechowuje zarówno wartości wariancji na przekątnej, jak i wartości kowariancji w komórkach niediagonalnych, jej prawdziwa nazwa to macierz wariancji-kowariancji, ale jest to dość rozwlekłe i dlatego powszechnie używa się terminu macierz kowariancji.

załóżmy poniższe dane początkowe

0.01,	0.10
0.02,	0.09
0.03,	0.10
0.04,	0.06
0.05,	0.06
0.06,	0.05
0.07,	0.05
0.08,	0.01
0.09,	0.02
0.10,	0.01

Rysunek 2.21. b14 dane wejściowe [89]

W klastrowaniu GMM istnieje jeden wielowymiarowy rozkład Gaussa i jeden współczynnik dla każdego skupienia. Zatem dla dwuwymiarowych danych demonstracyjnych z $k = 3$ skupieniami istnieją trzy rozkłady i trzy współczynniki. wartości pierwszego rozkładu Gaussa to:

Średnia [0] = [0.0550, 0.0550]

Macierz kowariancji [0] =

$$\begin{bmatrix} 0.0002 & -0.0001 \\ -0.0001 & 0.0000 \end{bmatrix}$$

Współczynnik [0] = 0.2017

wartości drugiego rozkładu Gaussa to:

Średnia [1] = [0.0550, 0.0550]

Macierz kowariancji [1] = $\begin{bmatrix} 0.0011 & -0.0011 \\ -0.0011 & 0.0011 \end{bmatrix}$

Współczynnik [1] = 0.5859

wartości trzeciego rozkładu Gaussa to:

Średnia [2] = [0.0550, 0.0550]

Macierz kowariancji [2] = $\begin{bmatrix} 0.0006 & -0.0011 \\ -0.0011 & 0.0019 \end{bmatrix}$

Współczynnik [2] = 0.2123

Te wartości rozkładu określają pseudoprawdopodobieństwa skupień dla danego sygnału wejściowego x . Dla pozycji danych [4] = [0,05, 0,06] pseudoprawdopodobieństwa skupień wynoszą [0,0087, 0,9296, 0,0617]. Oblicza się je w dwóch etapach w następujący sposób. Najpierw obliczana jest suma wartości rozkładu razy współczynniki:

Rozkład 0: ([0.05, 0.06]) · 0.2017

Rozkład 1: ([0.05, 0.06]) · 0.5859

Rozkład 2: ([0.05, 0.06]) · 0.2123

Obliczenia:

= 90.7419 · 0.2017 +

3328.6056 · 0.5859 +

2098.0352 · 0.2123

= 18.3071 + 1950.2563 + 129.4719

= 2098.0352

wartości współczynników są normalizowane tak, że sumują się do 1:

Pseudoprawdopodobieństwo[0] = $\frac{18.3071}{2098.0352} = 0.0087$

Pseudoprawdopodobieństwo[1] = $\frac{1950.2563}{2098.0352} = 0.9296$

Pseudoprawdopodobieństwo[2] = $\frac{129.4719}{2098.0352} = 0.0617$

to pseudoprawdopodobieństwa, że pozycja [0,05, 0,06] należy do skupienia 0, skupienia 1 i skupienia 2.

Tak więc, jeśli masz średnie, macierze kowariancji, współczynniki i sposób obliczenia rozkładu, stosunkowo łatwo jest obliczyć pseudoprawdopodobieństwa przynależności do klastrów.

[90][91][89][88]

2.4.2. Uczenie nadzorowane (supervised Learning)

Uczenie nadzorowane jest jednym z kluczowych obszarów uczenia maszynowego, który polega na trenowaniu modeli na podstawie danych, które mają już oznaczone etykiety lub wyniki. Celem jest nauczenie modelu przewidywania tych etykiet lub wyników dla nowych, nieoznaczonych danych. W tym rozdziale omówimy różne popularne algorytmy uczenia nadzorowanego, które znajdują zastosowanie w szerokim zakresie dziedzin, takich jak klasyfikacja i regresja.

Klasyfikacja i regresja, Załóżmy, że prognozujemy liczbę sprzedanych jednostek coli latem w określonym regionie. Wartość waha się pomiędzy pewnymi wartościami – powiedzmy 1 milion do 1,2 miliona jednostek tygodniowo. Zazwyczaj regresja jest sposobem prognozowania takie zmienne ciągłe. Z drugiej strony klasyfikacja lub przewidywanie przewiduje zdarzenia, które mają różne wyniki — na przykład to, czy dzień będzie słoneczny, czy deszczowy. Regresja liniowa jest typowym przykładem techniki ciągłego prognozowania zmiennych, podczas gdy regresja logistyczna jest typową techniką przewidywania zmiennych dyskretnych. Istnieje wiele innych technik, w tym drzewa decyzyjne, lasy losowe, sieci neuronowe i nie tylko, które mogą pomóc w przewidywaniu zarówno ciągłości, jak i wyników dyskretnych.

[92][74][93][94]

algorytmy uczenia nadzorowanego rozwiązujące problem regresji Przykłady algorytmów uczenia nadzorowanego rozwiązujący problem regresji:

- Support vector regression
- Neural Networks (NN)
- Recurrent Neural Networks (RNN)
- K-nearest neighbors regression
- regression tree
- linear regression
- polynomial regression
- Generalized regression neural networks
- Classification and regression trees (CART)
- ensemble methods

[93][28]

3. Metodologia

ma na celu przedstawienie dokładnego opisu metod i procedur, które zostały wykorzystane . pozwala innym zrozumieć, w jaki sposób przeprowadziłeś badanie i jakie kroki podjąłeś, aby uzyskać wyniki.

3.1. Gromadzenie i przygotowanie danych

opisuje, skąd pochodzą dane badawcze, jak je zebrano i jak zostały przygotowane do analizy. Może to obejmować proces zbierania próbek, źródła danych, metody gromadzenia informacji, a także kroki konieczne do oczyszczenia i przetworzenia danych, takie jak usuwanie błędów czy brakujących wartości.

3.2. Wybór algorytmu uczenia maszynowego:

wyjaśniam, dlaczego wybrałem określone algorytmy uczenia maszynowego do rozwiązania mojego problemu. Opisuje, jakie kryteria były brane pod uwagę przy wyborze algorytmów i dlaczego one były odpowiednie dla badania.

3.3. Rozwój modelu

opisuje kroki, które podjąłem w celu opracowania modelu uczenia maszynowego. To obejmuje wybór cech (feature selection), trenowanie modelu, dostrajanie parametrów, a także ewentualne eksperymenty i iteracje, które przeprowadziłeś, aby osiągnąć optymalne wyniki.

3.4. Metryki oceny modelu

wyjaśniam, jakie metryki i kryteria używam do oceny skuteczności modelu.

4. Wybrane algorytmy uczenia maszynowego

przedstawiam konkretne algorytmy i techniki uczenia maszynowego, które zostały użyte lub badane w projekcie. Każda podsekcja jest poświęcona innemu algorytmowi lub technice.

4.1. Regresja liniowa

Regresja liniowa to technika uczenia maszynowego wykorzystywana do modelowania zależności między jedną lub wieloma zmiennymi niezależnymi a zmienną zależną, która jest ciągła. Ta sekcja opisuje, w jaki sposób działa regresja liniowa, jakie są jej zalety i ograniczenia oraz w jakich sytuacjach może być użyteczna w analizie danych.

4.2. Drzewa decyzyjne

Drzewa decyzyjne są algorytmem, który pozwala na podejmowanie decyzji w oparciu o serię pytań i warunków logicznych. Ta sekcja wyjaśnia, jak drzewa decyzyjne są budowane, jakie są ich cechy i jak mogą być używane

4.3. Sieci neuronowe

Sieci neuronowe są algorytmami inspirowanymi budową mózgu, wykorzystywanymi do uczenia maszynowego. Ta sekcja opisuje podstawy sieci neuronowych, jakie są rodzaje sieci, jakie są zastosowania sieci neuronowych w problemach klasyfikacji i przewidywania.

4.4. Maszyny wektorów nośnych

Maszyny wektorów nośnych (SVM) to technika uczenia maszynowego, która jest wykorzystywana w zadaniach klasyfikacji i regresji. Wyjaśniasz, jak SVM działa, jakie są jego zalety, a także w jakich sytuacjach może być używan

4.5. Algorytmy klastrowania

Algorytmy klastrowania to techniki używane do grupowania danych na podstawie ich podobieństwa. Ta sekcja opisuje różne algorytmy klastrowania, jakie są ich zastosowania i jak mogą pomóc w analizie danych.

5. Studium przypadku

przedstawiam konkretne przypadki lub scenariusze, w których zastosowano algorytmy uczenia maszynowego w kontekście zarządzania łańcuchem dostaw. Każda podsekcja jest poświęcona innemu przypadkowi z wykorzystaniem tych algorytmów.

5.1. Studium przypadku 1: Optymalizacja zapasów

W tej sekcji przedstawiam konkretny scenariusz związany z zarządzaniem łańcuchem dostaw, gdzie wykorzystano algorytmy uczenia maszynowego do optymalizacji poziomu zapasów. Opisuję, jakie były cele tego studium przypadku, jakie dane zostały użyte do analizy oraz jakie algorytmy były stosowane w celu zoptymalizowania zarządzania zapasami.

5.2. Studium przypadku 2: Prognozowanie popytu

prezentuje konkretny przykład zastosowania algorytmów uczenia maszynowego do prognozowania popytu na produkty lub usługi w łańcuchu dostaw. Wyjaśniam, jakie były cele prognozowania popytu, jakie dane zostały użyte do tworzenia modeli prognozowania i jakie wyniki uzyskano w analizie.

5.3. Studium przypadku 3: Wybór dostawcy

Ta sekcja opisuje sytuację, w której algorytmy uczenia maszynowego zostały zastosowane do wyboru dostawcy w łańcuchu dostaw. Omawiam, jakie czynniki i kryteria były brane pod uwagę podczas procesu wyboru dostawcy, jakie dane były używane do analizy i jak algorytmy wspomagały w podjęciu decyzji.

5.4. Studium przypadku 4: Planowanie produkcji

W tej sekcji prezentuję przykład zastosowania algorytmów uczenia maszynowego do planowania produkcji w ramach zarządzania łańcuchem dostaw. Opisuję cele planowania produkcji, używane dane, a także wykorzystane algorytmy i techniki do optymalizacji procesów produkcyjnych.

6. Wyniki i dyskusja

prezentuje wyniki badań i analizuje ich znaczenie w kontekście zarządzania łańcuchem dostaw.

6.1. Porównanie wydajności algorytmów

W tej sekcji dokonuje porównania wyników osiągniętych przez różne algorytmy uczenia maszynowego, które zostały użyte w Twoich studiach przypadku (lub w planowaniu produkcji). To pozwala na ocenę, który z algorytmów sprawdził się najlepiej w konkretnej sytuacji. Moge użyć różnych metryk oceny, aby porównać wydajność

6.2. Interpretacja wyników

W tej podsekcji analizuje i interpretuje wyniki uzyskane w badaniach przypadku . Wyjaśniasz, co otrzymane wyniki oznaczają w kontekście Twojego badania. Czy wyniki potwierdzają lub kwestionują pierwotne hipotezy lub cele? W jaki sposób algorytmy uczenia maszynowego wpłynęły na efektywność zarządzania łańcuchem dostaw lub planowania produkcji?

6.3. Implikacje dla zarządzania łańcuchem dostaw

W tej sekcji rozważam, jakie praktyczne implikacje wyników badań mają dla zarządzania łańcuchem dostaw. Jakie wnioski można wyciągnąć na temat optymalizacji procesów, efektywności czy podejmowania decyzji w łańcuchu dostaw na podstawie uzyskanych wyników? Jakie praktyczne zastosowania lub zalecenia można sformułować na podstawie badań?

7. Wyzwania i przyszłe kierunki

analizuje istniejące wyzwania i przyszłe kierunki w dziedzinie zarządzania łańcuchem dostaw (Supply Chain Management - SCM) oraz wykorzystania technik uczenia maszynowego (ML) w tej dziedzinie.

7.1. Wyzwania we wdrażaniu ML w SCM

W tej części pracy koncentruje się na identyfikacji i analizie wyzwań związanych z wdrażaniem technik uczenia maszynowego w zarządzaniu łańcuchem dostaw. Omawiasz trudności związane z gromadzeniem, przetwarzaniem i analizą dużych zbiorów danych, akceptacją przez organizację, kosztami wdrożenia, a także możliwymi problemami związanymi z infrastrukturą i kompatybilnością z istniejącymi systemami SCM. Dostarczam także sugestii i rozważań dotyczących strategii radzenia sobie z tymi wyzwaniami.

7.2. Przyszłe kierunki badań

W tej części pracy rozważam przyszłe kierunki badań w dziedzinie zarządzania łańcuchem dostaw i zastosowań ML w SCM. Wskazujesz obszary, które mogą stać się głównymi tematami badawczymi w przyszłości, takie jak rozwijanie bardziej zaawansowanych modeli predykcyjnych, wykorzystanie danych czasu rzeczywistego, automatyzacja procesów SCM, czy też bardziej zaawansowane technologie śledzenia i monitorowania. Twoje refleksje mają na celu ukierunkowanie przyszłych prac badawczych i innowacji w tej dziedzinie.

7.3. Względy etyczne i dotyczące prywatności

W tej sekcji zajmujesz się kwestiami etycznymi i ochroną prywatności w kontekście wykorzystania ML w SCM. Rozważam potencjalne zagrożenia związane z danymi, takie jak naruszenia prywatności klientów lub pracowników oraz możliwe skutki uboczne stosowania algorytmów uczenia maszynowego. Omawiam także istniejące lub proponowane środki ostrożności i regulacje, które można wprowadzić w celu ochrony danych i zachowania etycznych standardów w zarządzaniu łańcuchem dostaw.

8. Wniosek

zakończenie pracy inżynierskiej i ma na celu podsumowanie kluczowych aspektów badania oraz przedstawienie ogólnych wniosków .

8.1. Podsumowanie ustaleń

W pierwszej części tej sekcji dokonuje podsumowania głównych ustaleń i wyników, które uzyskałem w trakcie swojego badania. To jest miejsce, gdzie można znaleźć najważniejsze informacje na temat badania w jednym miejscu. Staram się podkreślić, co udało Ci się osiągnąć i jakie kluczowe wyniki uzyskałeś.

8.2. Wkład

opisuje, jaki konkretny wkład praca wnosi do zarządzania łańcuchem dostaw i zastosowań uczenia maszynowego w SCM. Wyjaśniam, w jaki sposób badania rozwiązują istniejące problemy, rozwijają wiedzę lub otwierają nowe możliwości badawcze w tej dziedzinie. To jest miejsce, gdzie podkreślasz znaczenie pracy.

8.3. Zastosowania praktyczne

Omawiam praktyczne zastosowania wyników badania w rzeczywistym środowisku biznesowym. Wyjaśniam, w jaki sposób przedsiębiorstwa lub organizacje mogą wykorzystać ustalenia i techniki w praktyce. To pozwala zrozumieć, jakie korzyści mogą wyniknąć z mojej pracy dla przemysłu i sektora SCM.

8.4. Wnioski i uwagi końcowe

W tej ostatniej części pracy podkreślam główne wnioski i formułuję uwagi końcowe na temat całego procesu badawczego. podziękowania za wsparcie, podsumowania, dlaczego ta praca jest istotna i jakie znaczenie ma dla dziedziny zarządzania łańcuchem dostaw oraz zastosowań uczenia maszynowego w SCM.

Bibliografia

- [1] M. Weinke, "Machine Learning im Logistikmanagement –Entwicklung eines Gestaltungsansatzes zum Einsatz von ML-Anwendungen in logistischen Entscheidungsprozessen", prac. dokt., Technische Universität Berlin, 2023.
- [2] M. M. Robert Nowak, *Sztuczna inteligencja dla inżynierów : metody ogólne*. Oficyna Wydawnicza Politechniki Warszawskiej, 2022.
- [3] C. Wong, Z. X. Guo i S. Y. S. Leung, *Optimizing Decision Making in the Apparel Supply Chain Using Artificial Intelligence (AI)*. Woodhead Publishing, 2013.
- [4] A. Jóźwiak, "Metoda oceny jakości usług transportowych w łańcuchach dostaw z wykorzystaniem algorytmów sztucznej inteligencji / Quality assessment of transport services in supply chains using artificial intelligence algorithms", prac. dokt., Politechnika Warszawska, 2017.
- [5] Wikipedia, *Zarządzanie łańcuchem dostaw*, Dostęp zdalny (14.03.2019): https://pl.wikipedia.org/wiki/Zarządzanie_łańcuchem_dostaw, 2023.
- [6] E. Zarządzania, *Zarządzanie łańcuchem dostaw*, Dostęp zdalny (20.09.2019): https://mfiles.pl/pl/index.php/Łańcuch_dostaw, 2023.
- [7] SAP, *Na czym polega zarządzanie łańcuchem dostaw (SCM)?*, Dostęp zdalny (20.09.2023): <https://www.sap.com/poland/products/scm/what-is-supply-chain-management.html>, 2023.
- [8] W. przewodnik logistyczny, *Co to jest łańcuch dostaw? Przewodnik*, Dostęp zdalny (20.09.2019): <https://wdx.pl/2021/10/28/co-to-jest-lancuch-dostaw-przewodnik/>, 2021.
- [9] Wikipedia, *Supply chain management*, Dostęp zdalny (14.03.2019): https://en.wikipedia.org/wiki/Supply_chain_management, 2023.
- [10] O. supply chain management, *Czym jest sztuczna inteligencja*, Dostęp zdalny (20.09.2023): <https://www.oracle.com/pl/scm/what-is-supply-chain-management/>, 2023.
- [11] S. R. P. /. sztucznej inteligencji, *Co to jest łańcuch dostaw? Przewodnik*, Dostęp zdalny (20.09.2023): <https://www.gov.pl/web/ai/czym-jest-sztuczna-inteligencja2>, 2023.
- [12] I. H.-L. E. G. O. A. I. S. U. B. T. E. COMMISSION, "A DEFINITION OF AI:MAIN CAPABILITIES AND DISCIPLINES", European Foundation for the Improvement of Living i Working Conditions, spraw. tech., 2019.
- [13] P. Statystyczne, *Szeregi czasowe*, Dostęp zdalny (14.03.2019): <https://pogotowiestatystyczne.pl/slowniki/szereg-czasowy/>, 2023.
- [14] Wikipedia, *Time series*, Dostęp zdalny (14.03.2019): https://en.wikipedia.org/wiki/Time_series, 2023.
- [15] S. R. Vasudev, "Demand forecasting using statistical and machine learning algorithms", prac. mag., 2018.

- [16] D. inż. Sebastian Skoczypiec, *Metody prognozowania: Szeregi czasowe*, Dostęp zdalny (14.03.2019): <https://m6.pk.edu.pl/>, 2009.
- [17] wikipedia, *Autoregresja*, Dostęp zdalny (20.09.2019): <https://pl.wikipedia.org/wiki/Autoregresja>, 2023.
- [18] wikipedia, *Model AR*, Dostęp zdalny (20.09.2019): [zrodlo:https://pl.wikipedia.org/wiki/Model_AR](https://pl.wikipedia.org/wiki/Model_AR), 2023.
- [19] pl.economy-pedia.com, *Model AR (1) - Co to jest, definicja i pojęcie*, Dostęp zdalny (20.09.2019): <https://pl.economy-pedia.com/11039603-model-ar-1>, 2023.
- [20] K. Muszyński, “Metoda sztucznych sieci neuronowych w prognozowaniu bieżącym zapotrzebowania na wodę w Krakowie”, prac. dokt., Politechnika Krakowska, 2012.
- [21] D. N. P. Sumaiya Farzana Ga, “Machine Learning in Demand Forecasting - A Review”, 2020.
- [22] wikipedia, *Metoda naiwna*, Dostęp zdalny (20.09.2019): [zrodlo:https://pl.wikipedia.org/wiki/Metoda_naiwna](https://pl.wikipedia.org/wiki/Metoda_naiwna), 2023.
- [23] Wikipedia, *Wyglądanie wykładnicze*, Dostęp zdalny (14.03.2019): https://pl.wikipedia.org/wiki/Wygladanie_wykładnicze, 2023.
- [24] www.ekonometria.4me.pl, *Model liniowy Holta Prognozowanie i symulacje*, Dostęp zdalny (14.03.2019): <http://www.ekonometria.4me.pl/model-Holta.htm>, 2023.
- [25] R. Szostek, “UOGÓLNIENIE MODEL HOLTA NA PRZYKŁADZIE PROGNOZOWANIA LICZBY PASAŻERÓW W TRANSPORCIE LOTNICZYM W POLSCE”, *Ekonometria / Uniwersytet Ekonomiczny we Wrocławiu, spraw. tech.*, 2012.
- [26] excelraport.pl, *Prognozowanie w Excelu Modelem Holta-Wintersa*, Dostęp zdalny (20.09.2023): <https://excelraport.pl/index.php/2018/03/05/prognozowanie-w-excelu-modelem-holta-wintersa/>, 2018.
- [27] ekonometria.4me.pl, *Metoda Wintersa addytywna i multiplikatywna*, Dostęp zdalny (20.09.2023): <http://www.ekonometria.4me.pl/metoda-wintersa.htm>, 2023.
- [28] P. modeli uczenia maszynowego wraz z przykładami zastosowania, *Co to jest łańcuch dostaw? Przewodnik*, Dostęp zdalny (20.09.2023): <https://www.gov.pl/web/popcwsparcie/podzial-modeli-uczenia-maszynowego-wraz-z-przykladami-zastosowania>, 2023.
- [29] A. I. Sreekanth Menon i G. Rajeev Ranjan data science leader, *The evolution of forecasting techniques: Traditional versus machine learning methods*, Dostęp zdalny (20.09.2023): <https://www.genpact.com/insight/the-evolution-of-forecasting-techniques-traditional-versus-machine-learning-methods>, 2023.
- [30] K. G. Nowadly i S. Jung, “Using Machine Learning Approaches to Improve Long-Range Demand Forecasting”, prac. mag., MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 2020.
- [31] R. V. Real Carbonneau Kevin Laframboise, “Application of machine learning techniques for supply chain demand forecasting”, *European Journal of Operational Research* 184 (2008) 1140–1154, 2006.

- [32] M. R. V. K. A. S. Maryam Zohdi, "Demand forecasting based machine learning algorithms on customer information: an applied approach", *International Journal of Information Technology* · February 2022, 2022.
- [33] wikipedia, *k-nearest neighbors algorithm*, Dostęp zdalny (20.09.2023): https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm, 2023.
- [34] A. N. H. Z. Z. L. Xiaodan Zhu, "Demand Forecasting with Supply-Chain Information and Machine Learning: Evidence in the Pharmaceutical Industry", *Production and Operations Management Society Vol. 30, No. 9, September 2021, pp. 3231–3252*, 2021.
- [35] R. Tugay i S. G. Oguducu, "Demand Prediction using Machine Learning Methods and Stacked Generalization", Conference: DATA'17At: Madrid, 2017.
- [36] R. Carbonneau, R. Vahidov i K. Laframboise, "Machine learning-Based Demand forecasting in supply chains", *International Journal of Intelligent Information Technologies*, Volume 3, Issue 4, 2007.
- [37] J. Shahrabi, 2. Mousavi i M. Heydar, "Supply Chain Demand Forecasting; A Comparison of Machine Learning Techniques and Traditional Methods", *Journal of Applied Sciences* 9 (3): 521 -527, 2009, 2009.
- [38] E. zarządzania, *Zarządzanie zapasami*, Dostęp zdalny (20.09.2023): https://mfiles.pl/pl/index.php/Zarzadzanie_zapasami, 2023.
- [39] P. K. B, P. J, P. Kumar i P. G, "Inventory Management using Machine Learning", *International Journal of Engineering Research Technology (IJERT)* ISSN: 2278-0181 Vol. 9 Issue 06, June-2020, 2020.
- [40] R. D. Thomas Hanne, *Computational Intelligence in Logistics and Supply Chain Management*. Springer, 2017.
- [41] linnworks.com, *AI inventory management: 9 ways AI can streamline inventory control*, Dostęp zdalny (20.09.2023): <https://www.linnworks.com/blog/ai-inventory-management/>, 2023.
- [42] cdp.com, *AI in Procurement: How Machine Learning is Changing the Game*, Dostęp zdalny (20.09.2023): <https://cdp.com/articles/ai-procurement-machine-learning/>, 2023.
- [43] M. Vahermo, "Exploring Machine Learning to Improve Procurement and Purchasing Processes", *prac. mag., UNIVERSITY OF VAASA School of Technology i Innovation*, 2023.
- [44] H. J. Wahedi, M. Heltoft, G. J. Christophersen, T. Severinsen, S. Saha i I. E. Nielsen, "Forecasting and Inventory Planning: An Empirical Investigation of Classical and Machine Learning Approaches for Svanehøj's Future Software Consolidation", *Applied Science* 2023, 13(15), 8581; <https://doi.org/10.3390/app13158581>, 2021.
- [45] wikipedia, *Support vector machine*, Dostęp zdalny (20.09.2023): https://en.wikipedia.org/wiki/Support_vector_machine, 2023.
- [46] A. Sethi, *Support Vector Regression Tutorial for Machine Learning*, Dostęp zdalny (20.09.2023): <https://www.analyticsvidhya.com/blog/2020/03/support->

- vector-regression-tutorial-for-machine-learning/introduction-to-support-vector-regression-svr, 2023.
- [47] intellipaas.com, *What is Q-Learning? Beginners Guide*, Dostęp zdalny (20.09.2023): <https://intellipaas.com/blog/q-learning/>, 2023.
- [48] E. zarządzania, *Planowanie produkcji*, Dostęp zdalny (20.09.2023): https://mfiles.pl/pl/index.php/Planowanie_produkcji, 2023.
- [49] planet together, *Objectives of Production Planning and Control*, Dostęp zdalny (20.09.2023): <https://www.planettogether.com/blog/objectives-of-production-planning-and-control>, 2023.
- [50] planet together, *Five Types of Production Planning*, Dostęp zdalny (20.09.2023): <https://www.planettogether.com/blog/five-types-of-production-planning>, 2023.
- [51] deskera, *Production Planning and Scheduling: The Complete Guide*, Dostęp zdalny (20.09.2023): <https://www.deskera.com/blog/production-planning-scheduling/>, 2023.
- [52] wikipedia, *Master production schedule*, Dostęp zdalny (20.09.2023): https://en.wikipedia.org/wiki/Master_production_schedule, 2023.
- [53] wikipedia, *Material requirements planning*, Dostęp zdalny (20.09.2023): https://en.wikipedia.org/wiki/Material_requirements_planning, 2023.
- [54] wikipedia, *capacity planning*, Dostęp zdalny (20.09.2023): https://en.wikipedia.org/wiki/Capacity_planning, 2023.
- [55] T. Ryback, L. Lingitz, A. G. V. Gallina i D. Gyulai, "Improving the Planning Quality in Production Planning and Control with Machine Learning", 16th IMEKO TC10 Conference "Testing, Diagnostics Inspection as a comprehensive value chain for Quality Safety Berlin, Germany, on September 3-4, 2019, 2019.
- [56] F. Guo, Y. Li, A. Liu i Z. Liu, "A Reinforcement Learning Method to Scheduling Problem of Steel Production Process", *Journal of Physics: Conference Series* 1486 072035, 2020.
- [57] wikipedia, *Uczenie przez wzmacnianie*, Dostęp zdalny (20.09.2023): https://pl.wikipedia.org/wiki/Uczenie_przez_wzmacnianie, 2023.
- [58] wikipedia, *Reinforcement learning*, Dostęp zdalny (20.09.2023): https://en.wikipedia.org/wiki/Reinforcement_learning, 2023.
- [59] wikipedia, *Markov decision process*, Dostęp zdalny (20.09.2023): https://en.wikipedia.org/wiki/Markov_decision_process, 2023.
- [60] J. K. Edward Nawarecki, "Agentowy model systemu logistycznego", *automatyka 2009 tom 13 zeszyt 2*, 2009.
- [61] G. Faustman, "Application of machine learning in production scheduling", TU Wien, 2019.
- [62] E. H. Alberto Garrea Mari Carmen Ruizb, "Application of Machine Learning to support production planning of a food industry in the context of waste generation under uncertainty", *Operations Research Perspectives Volume 7, 2020, 100147*, 2020.

- [63] M. D. Swetha Govindaiah, "Applying reinforcement learning to plan manufacturing material handling", *Discover Artificial Intelligence (2021) 1:8*, 2021.
- [64] matlab, *What is Model Predictive Control?*, Dostęp zdalny (20.09.2023): <https://www.mathworks.com/help/mpc/gs/what-is-mpc.html>, 2023.
- [65] Y. Chen, Y. Zhou i Y. Zhang, "Machine Learning-Based Model Predictive Control for Collaborative Production Planning Problem with Unknown Information", *Electronics 2021, 10, 1818*. <https://doi.org/10.3390/electronics10151818>, 2021.
- [66] P. C. S. specjalista i web analityk w Greenlogic., *Ensemble learning. Czym jest i na czym polega?*, Dostęp zdalny (20.09.2023): <https://geek.justjoin.it/ensemble-learning-czym-czym-polega/>, 2023.
- [67] J. P. U. Cadavid, S. Lamouri, B. Grabot, R. Pellerin i A. Fortin, "Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0", *Journal of Intelligent Manufacturing, 2020, 31 (6), pp.1531-1558.*, 2020.
- [68] A. Estesó, D. Peidro, J. Mula i M. Díaz-Madroñero, "Reinforcement learning applied to production planning and control", *INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH 2023, VOL. 61, NO. 16, 5772-57899*, 2023.
- [69] H. W. Gustaf Ehn, "REINFORCEMENT LEARNING FOR PLANNING OF A SIMULATED PRODUCTION LINE", prac. mag., Lund University, Faculty of Engineering Centre for Mathematical Sciences Mathematics, 2018.
- [70] M. Szeliga, *Praktyczne machine learning*. Helion, 2019.
- [71] G. S. ; D. S. ; P. R. Valentino Zocca, *Deep learning : uczenie głębokie z językiem Python : sztuczna inteligencja i sieci neuronowe*. Helion, 2018.
- [72] H. de Ponteves, *Sztuczna inteligencja : błyskawiczne wprowadzenie do uczenia maszynowego, uczenia ze wzmocnieniem i uczenia głębokiego*. Helion, 2021.
- [73] openai, *Welcome to Spinning Up in Deep RL!*, Dostęp zdalny (14.03.2019): <https://spinningup.openai.com/en/latest/index.html>, 2023.
- [74] A. N. Morken, "Using Machine Learning for Improved Production Planning in the Concrete Supplier Business", prac. mag., Norwegian University of Science, Technology: Department of Mechanical i Industrial Engineering, 2021.
- [75] M. McGregor, *8 Clustering Algorithms in Machine Learning that All Data Scientists Should Know*, Dostęp zdalny (20.09.2023): <https://machinelearningmastery.com/clustering-algorithms-with-python/>, 2020.
- [76] datascience.eu, *Klastrowanie*, Dostęp zdalny (20.09.2023): <https://datascience.eu/pl/matematyka-i-statystyka/tworzenie-klastrow/>, 2020.
- [77] K. Rządca, "Algorytmy grupowania danych", prac. mag., Politechnika Warszawska Wydział Elektroniki i Technik Informacyjnych Instytut Informatyki, 2004.
- [78] naukowiec.org, *Rozkład normalny, Rozkład Gaussa - opis*, Dostęp zdalny (20.09.2023): https://www.naukowiec.org/wiedza/statystyka/rozklad-normalny-rozklad-gaussa_710.html, 2023.

- [79] J. Brownlee, *10 Clustering Algorithms With Python*, Dostęp zdalny (20.09.2023): <https://www.freecodecamp.org/news/8-clustering-algorithms-in-machine-learning-that-all-data-scientists-should-know/>, 2020.
- [80] T. Yenigün, *An In-depth Examination and Demonstration in Python The Mechanics of Affinity Propagation Clustering*, Dostęp zdalny (20.09.2023): <https://python.plainenglish.io/the-mechanics-of-affinity-propagation-clustering-eb199cc7a7c2>, 2023.
- [81] T. Gorecki, *analiza danych*, Dostęp zdalny (20.09.2023): , 2023.
- [82] C. Maklin, *Affinity Propagation Algorithm Explained*, Dostęp zdalny (20.09.2023): <https://towardsdatascience.com/unsupervised-machine-learning-affinity-propagation-algorithm-explained-d1fef85f22c8>, 2019.
- [83] Praneethsanthosh, *Clustering Algorithm- Affinity propagation*, Dostęp zdalny (20.09.2023): <https://praneeth88.medium.com/clustering-algorithm-affinity-propagation-5d7e2a44adef>, 2021.
- [84] Aditya, *Agglomerative Clustering Numerical Example, Advantages and Disadvantages*, Dostęp zdalny (20.09.2023): <https://codinginfinite.com/agglomerative-clustering-numerical-example-advantages-and-disadvantages/>, 2022.
- [85] E. C. of Science, *10.2 - Example: Agglomerative Hierarchical Clustering*, Dostęp zdalny (20.09.2023): <https://online.stat.psu.edu/stat555/node/86/>, 2023.
- [86] T. H. ., *DBSCAN — Make density-based clusters by hand*, Dostęp zdalny (20.09.2023): <https://towardsdatascience.com/dbscan-make-density-based-clusters-by-hand-2689dc335120>, 2020.
- [87] P. Gumowski, “Zastosowanie algorytmu DBSCAN do grupowania danych rozproszonych .”, prac. mag., Politechnika Warszawska Wydział Elektroniki i Technik Informacyjnych Instytut Informatyk, 2015.
- [88] G. Bonaccorso, *Mastering Machine Learning Algorithms*. Packt Publishing Ltd., 2018.
- [89] J. McCaffrey, *Gaussian Mixture Model Data Clustering from Scratch Using C*, Dostęp zdalny (20.09.2023): <https://visualstudiomagazine.com/articles/2023/11/01/gaussian-mixture-model-data-clustering.aspx>, 2023.
- [90] data flair, *Gaussian Mixture Model with Case Study – A Survival Guide for Beginners*, Dostęp zdalny (20.09.2023): <https://data-flair.training/blogs/gaussian-mixture-model/>, 2023.
- [91] baeldung, *Gaussian Mixture Models*, Dostęp zdalny (20.09.2023): <https://www.baeldung.com/cs/gaussian-mixture-models>, 2023.
- [92] V. K. Ayyadevara, *Pro Machine Learning Algorithms A Hands-On Approach to Implementing Algorithms in Python and R*. Apress, 2018.
- [93] D. S. Hannah Wenzel i S. Sardesai, “A Literature Review on Machine Learning in Supply Chain Management”, *Artificial Intelligence and Digital Transformation in Supply Chain Management*, 2019.
- [94] W.-m. lee, *Python machine learning*. John Wiley Sons, 2019.

Wykaz symboli i skrótów

EiTI – Wydział Elektroniki i Technik Informatycznych

PW – Politechnika Warszawska

WEIRD – ang. *Western, Educated, Industrialized, Rich and Democratic*

Spis rysunków

2.1	Obszary zastosowań sztucznej inteligencji[11]	17
2.2	s1a Modele szeregów czasowych[16]	21
2.3	k1 Grupowanie K-średnich[75]	56
2.4	k2 Grupowanie K-średnich przypisanie danych[75]	56
2.5	k4 Grupowanie K-średnich,dane przypisane do klastrów[75]	57
2.6	k6 Grupowanie K-średnich,dane przypisane do klastrów[75]	57
2.7	k7 Grupowanie K-średnich,dane przypisane do klastrów[75]	58
2.8	b1 Grupowanie affinity propagation,wskazanie przykładów (exemplars)[80]	59
2.9	b2 Grupowanie affinity propagation,wskazanie klastrów[80]	59
2.10	b3 Tabela z danymi wejściowymi[81]	60
2.11	b4 macierz podobieństw[81]	61
2.12	b5 macierz odpowiedzialności R[81]	62
2.13	b6 macierz dostępności A[81]	62
2.14	b7 macierz kryterium C[81]	62
2.15	b8 macierz odległości[84]	63
2.16	b9 dendrogram[84]	64
2.17	b10 dane wejściowe[86]	65
2.18	b11 eps 0.3 dla punktu (1.5,2.5) [86]	66
2.19	b12 punkty graniczne, centralne i odstające dla eps=6 i minPts=4 [86]	67
2.20	b13 macierz kowariancji[89]	68
2.21	b14 dane wejściowe[89]	69

Spis tabel

Spis załączników

1.	algorytmy klasteryzacji	86
2.	Nazwa załącznika 2	114

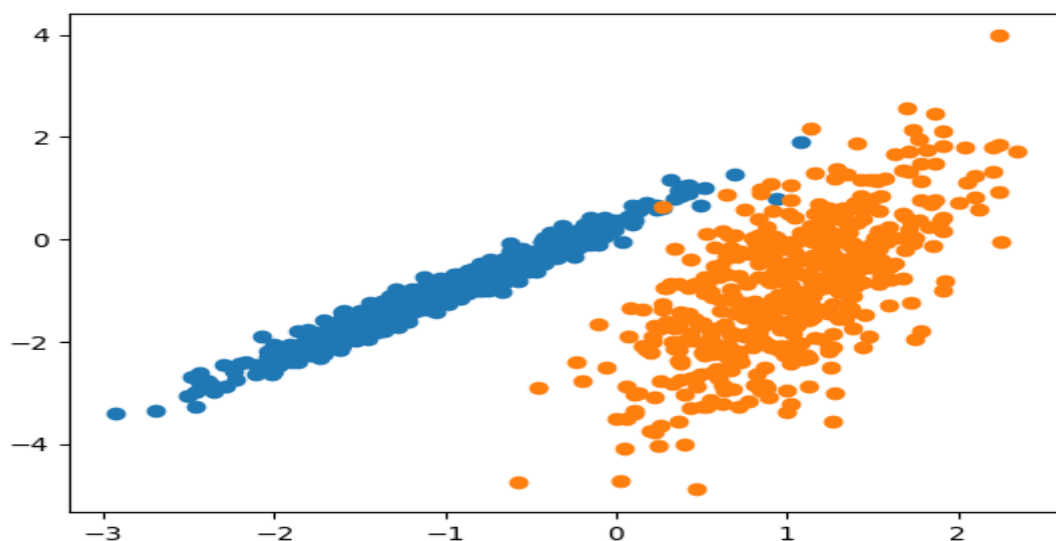
Załącznik 1. algorytmy klasteryzacji

Przedstawiam algorytmy które służą do klasteryzacji danych
Podstawowa klasyfikacja (w języku python):

Przykład klasyfikacji języku python

```
#importuje funkcję where z biblioteki NumPy, która jest używana do uzyskiwania indeksów
elementów spełniających określone warunki.
#importuje funkcję make_classification z biblioteki scikit-learn, która pozwala na
generowanie zestawów danych do klasyfikacji.
#importuje moduł pyplot z biblioteki Matplotlib, który umożliwia tworzenie wykresów.
from numpy import where
from sklearn.datasets import make_classification
from matplotlib import pyplot

# define datasettworzy zestaw danych do klasyfikacji. Parametry funkcji make_classification
określają właściwości tego zestawu danych,
# takie jak liczba próbek (n_samples), liczba cech (n_features), liczba informacyjnych
cech (n_informative), brak redundantnych cech (n_redundant),
# liczba klastrów na klasę (n_clusters_per_class) itp. Parametr random_state=4 kontroluje
generowanie zestawu danych,
# co oznacza, że za każdym razem, gdy ustawiony jest na tę samą wartość (tutaj 4), zostanie
wygenerowany ten sam zestaw danych.
X, y = make_classification(n_samples=1000, n_features=2, n_informative=2, n_redundant=0,
n_clusters_per_class=1, random_state=4)
#pętlę for, która będzie iterować po dwóch wartościach klas (0 i 1), które są dostępne w
zestawie danych.
for class_value in range(2):
    # używa funkcji where do uzyskania indeksów próbek, które należą do aktualnie
rozpatrywanej klasy (class_value).
    # Indeksy te są przechowywane w zmiennej row_ix.
    row_ix = where(y == class_value)
    # tworzy wykres punktowy za pomocą funkcji scatter z modułu pyplot. Wykres ten
przedstawia próbki z aktualnej klasy.
    # Współrzędne x i y tych próbek są pobierane z macierzy X na podstawie indeksów
zapisanych w row_ix.
    pyplot.scatter(X[row_ix, 0], X[row_ix, 1])
# wyświetla wykres punktowy
pyplot.show()
```



klasteryzacja , model Affinity Propagation (python).Ten kod wykonuje klastrowanie przy użyciu algorytmu Affinity Propagation na zestawie danych i wizualizuje wyniki na wykresie punktowym.

```
# importuje funkcję unique z biblioteki NumPy, która jest używana do uzyskania unikalnych elementów w tablicy.
# importuje funkcję where z biblioteki NumPy, która jest używana do uzyskiwania indeksów elementów spełniających określone warunki.
# importuje funkcję make_classification z biblioteki scikit-learn, która pozwala na generowanie zestawów danych do klasyfikacji.
# importuje model AffinityPropagation z biblioteki scikit-learn, który jest używany do przeprowadzenia klastrowania przy użyciu algorytmu
#Affinity Propagation.
#importuje moduł pyplot z biblioteki Matplotlib, który umożliwia tworzenie wykresów.
from numpy import unique
from numpy import where
from sklearn.datasets import make_classification
from sklearn.cluster import AffinityPropagation
from matplotlib import pyplot
# tworzy zestaw danych do klasyfikacji. Oto opis parametrów funkcji make_classification:
n_samples=1000:
# Określa liczbę próbek w zestawie danych, które wynosi 1000.
# n_features=2: Określa liczbę cech w zestawie danych, która wynosi 2.
# n_informative=2: Określa liczbę informacyjnych cech, które mają wpływ na rozróżnianie klas, co wynosi 2.
# n_redundant=0: Określa liczbę redundantnych cech, które nie przynoszą dodatkowej informacji i wynosi 0.
# n_clusters_per_class=1: Określa liczbę klastrów na każdą klasę w zestawie danych, co wynosi 1.
# random state=4: Parametr kontroluje reprodukowalność procesu generowania danych losowych i ustawiony na 4 oznacza,
# że proces ten będzie taki sam za każdym razem, gdy używamy tej samej wartości random state.
X, _ = make_classification(n_samples=1000, n_features=2, n_informative=2, n_redundant=0, n_clusters_per_class=1, random_state=4)
# tworzy model AffinityPropagation z określonym parametrem damping, który wpływa na tempo zbieżności algorytmu klastrowania.
# parametr damping jest używany w algorytmie Affinity Propagation w celu kontrolowania tempa zbieżności algorytmu klastrowania
# W każdej iteracji algorytmu, próbki komunikują się ze sobą, a "afinność" między nimi jest obliczana. Parametr damping wpływa na tempo,
# z jakim te komunikaty się rozprzestrzeniają i jak szybko algorytm zbiega do ostatecznych klastrów. Gdy damping jest bliski 1 (np. 0.9)
# to komunikaty rozprzestrzeniają się szybko, co może prowadzić do szybkiego zbieżenia algorytmu. To oznacza, że algorytm może wykryć klastry
# stosunkowo szybko, ale istnieje ryzyko, że wykryte klastry mogą być mniej stabilne.Gdy damping jest bliski 0 (np. 0.5),
# to komunikaty rozprzestrzeniają się wolniej, co może prowadzić do bardziej stabilnych klastrów. Jednak zbieżność algorytmu może być wolniejsza,
# a algorytm może wymagać większej liczby iteracji.X to zestaw danych, dla którego chcemy dokonać klastrowania model.predict(X)
# jest wywołaniem metody predict na wytrenowanym modelu model. Ta metoda przyjmuje zestaw danych X i przypisuje każdej próbce etykietę klastra
# na podstawie wyników klastrowania, które zostały uzyskane podczas procesu treningu modelu.
model = AffinityPropagation(damping=0.9)
```



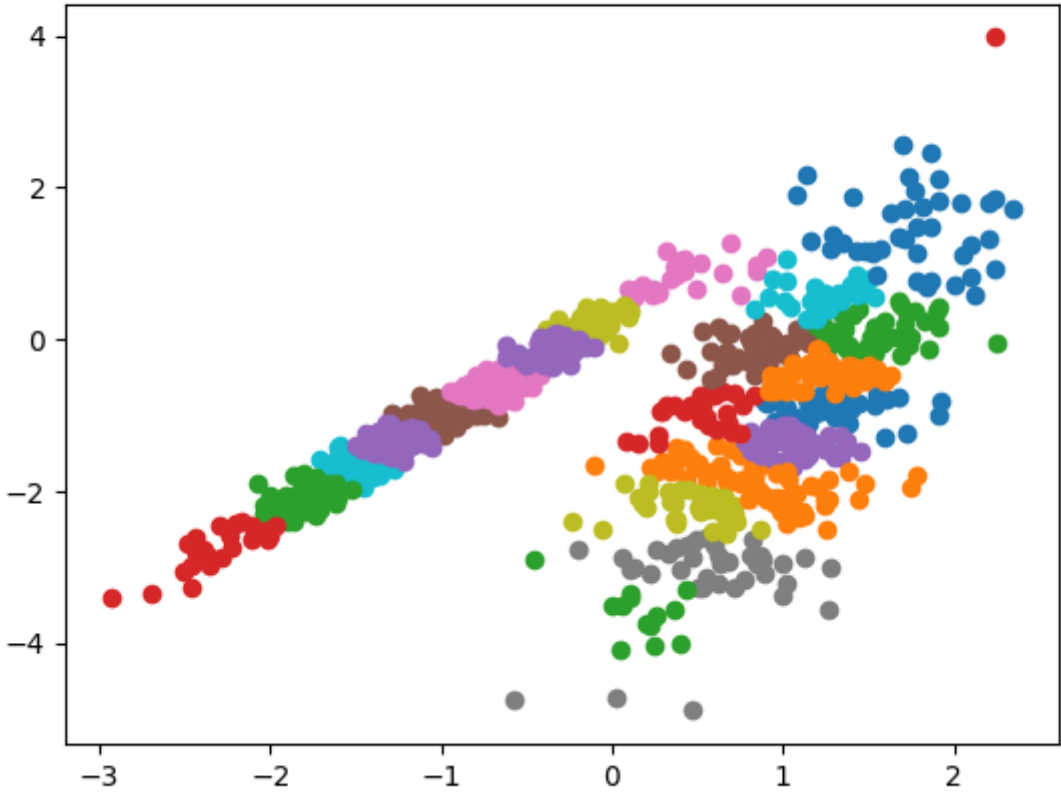
```

# dopasowuje model AffinityPropagation do danych X.
model.fit(X)
# przypisuje każdej próbce etykietę klastra na podstawie wyników klastrowania. Jest używana
do przewidywania przynależności klastrów
# dla każdej próbki w zestawie danych X za pomocą wytrenowanego modelu klastrowania
AffinityPropagation.model to wytrenowany model klastrowania
# AffinityPropagation, który został wcześniej zdefiniowany w kodzie za pomocą linii model
= AffinityPropagation(damping=0.9).
# Ten model zawiera informacje o klastrach, które zostały wykryte w danych
treningowych. Wynik tego wyrażenia, yhat, to tablica zawierająca
# etykiety klastrów przypisane do każdej próbki w zestawie danych X. Każda próbka w
zestawie danych jest przyporządkowana do jednego z klastrów.
yhat = model.predict(X)
# uzyskuje unikalne etykiety klastrów na podstawie wyników klastrowania. unique(yhat) to
wywołanie funkcji unique z biblioteki NumPy na tablicy yhat.
# Funkcja ta jest używana do znalezienia unikalnych wartości w tablicy lub wektorze. W
kontekście tego kodu, yhat zawiera etykiety klastrów przypisane
# do próbek w zestawie danych po przeprowadzeniu klastrowania. clusters to zmienna, do
której przypisywane są unikalne etykiety klastrów znalezione
# w wynikach klastrowania. Każda etykieta odpowiada jednemu klastrze, a clusters jest
tablicą zawierającą te unikalne etykiety. Ten krok jest przydatny,
# gdy chcemy uzyskać listę wszystkich klastrów znalezionych w danych po klastrowaniu. W tym
konkretnym przypadku, wynik clusters jest później
# wykorzystywany w pętli for cluster in clusters:, aby iterować przez unikalne klastry i
przeprowadzać operacje na danych dla każdego klastra osobno,
# na przykład tworzyć wykresy lub analizować charakterystyki klastrów.
clusters = unique(yhat)
# rozpoczyna pętlę, która będzie iterować przez unikalne klastry znalezione w wynikach
klastrowania.
for cluster in clusters:
    # uzyskuje indeksy próbek, które należą do aktualnie rozpatrywanego klastra.
    row_ix = where(yhat == cluster)
    # tworzy wykres punktowy, na którym wyświetlane są próbki należące do aktualnego klastra.
Współrzędne x i y tych próbek są pobierane z macierzy X
    # na podstawie indeksów zapisanych w row_ix. X[row_ix, 0] to wyrażenie, które pobiera
współrzędną X (pierwszą cechę) dla próbek należących
    # do aktualnie rozpatrywanego klastra. row_ix zawiera indeksy tych próbek. Wyrażenie to
tworzy wektor wartości współrzędnych X dla danego klastra.
    # X[row_ix, 1] to analogicznie druga cecha, która reprezentuje współrzędną Y na wykresie.
Również tworzy wektor wartości współrzędnych Y dla danego
    # klastra. X[row_ix, 0] i X[row_ix, 1] razem określają położenie punktów na wykresie
punktowym. Każda próbka w danym klastrze jest reprezentowana
    # jako punkt o współrzędnych (X[row_ix, 0], X[row_ix, 1]) na wykresie. Oznacza to, że dla
każdej próbki w klastrze zostaje narysowany jeden punkt
    # na wykresie. Wynik jest taki, że każdy klaster zostaje reprezentowany na wykresie
punktowym przez różne kolory lub symbole, a punkty w obrębie
    # jednego klastra są blisko siebie
    pyplot.scatter(X[row_ix, 0], X[row_ix, 1])
# pokazuje wykres
pyplot.show()

```

ponizej wykres wygenerowany przez kod:

Figure 1



klasteryzacja , model Agglomerative Clustering (python). Ten kod jest przykładem zastosowania algorytmu aglomeracyjnego (agglomerative clustering) do grupowania danych.

```
# Importuje funkcję unique z biblioteki NumPy, która zostanie użyta do znalezienia
unikalnych wartości w wektorze.
# Importuje funkcję where z biblioteki NumPy, która zostanie użyta do znalezienia indeksów,
które spełniają określone warunki.
# importuje funkcję make_classification z biblioteki Scikit-Learn, która zostanie użyta do
wygenerowania zestawu danych do eksperymentów z grupowaniem.
# Importuje klasę AgglomerativeClustering z biblioteki Scikit-Learn, która reprezentuje
algorytm aglomeracyjnego grupowania.
# Importuje moduł pyplot z biblioteki Matplotlib, który zostanie użyty do rysowania
wykresów.
from numpy import unique
from numpy import where
from sklearn.datasets import make_classification
from sklearn.cluster import AgglomerativeClustering
from matplotlib import pyplot
# Generowanie zestawu danych z 2 cechami i 1000 próbkami za pomocą funkcji
make_classification. Parametry funkcji make_classification:
# - n_samples: Liczba próbek (1000).
# - n_features: Liczba cech (2).
# - n_informative: Liczba informatywnych cech (2) - cechy te wpływają na separację klas.
# - n_redundant: Liczba nadmiarowych cech (0) - cechy te są losowo wygenerowane i nie
wpływają na separację klas.
# - n_clusters_per_class: Liczba klastrów na klasę (1) - oznacza, że każda klasa jest
jednym skupiskiem danych.
# - random_state: Ziarno generatora liczb losowych (ustawione na 4 dla powtarzalności
eksperymentu).
X, _ = make_classification(n_samples=1000, n_features=2, n_informative=2, n_redundant=0,
n_clusters_per_class=1, random_state=4)
# Inicjalizacja modelu AgglomerativeClustering z dwoma klastrami. W tym wierszu
tworzymy instancję algorytmu grupowania aglomeracyjnego,
# który będzie próbował podzielić dane na 2 klastry. Parametr n_clusters ustawiony na 2
określa liczbę klastrów, które algorytm spróbuje znaleźć.
model = AgglomerativeClustering(n_clusters=2)
# Dopasowanie modelu do danych i przewidywanie przynależności próbek do klastrów,
zapisując wyniki w yhat. Metoda fit_predict wykonuje proces
# grupowania na danych X i zwraca wektor yhat, w którym każda próbka jest przypisana do
jednego z klastrów.
yhat = model.fit_predict(X)
# Znalezienie unikalnych klastrów w wynikach przewidywania yhat. Funkcja unique z
biblioteki NumPy pozwala na znalezienie unikalnych wartości
# w wektorze, co oznacza, że ta linia kodu znajduje różne klastry, które zostały znalezione
w danych.
clusters = unique(yhat)
# Pętla iterująca przez unikalne klastry.
for cluster in clusters:
    # Dla każdego klastra znajdź indeksy próbek należących do tego klastra. Funkcja where
z biblioteki NumPy znajduje indeksy w yhat,
    # które odpowiadają obecnie badanemu klastrze.
    row_ix = where(yhat == cluster)
    # Tworzenie wykresu punktowego, na którym wyświetlane są próbki z danego klastra. Dla
każdego klastra tworzony jest wykres punktowy,
```

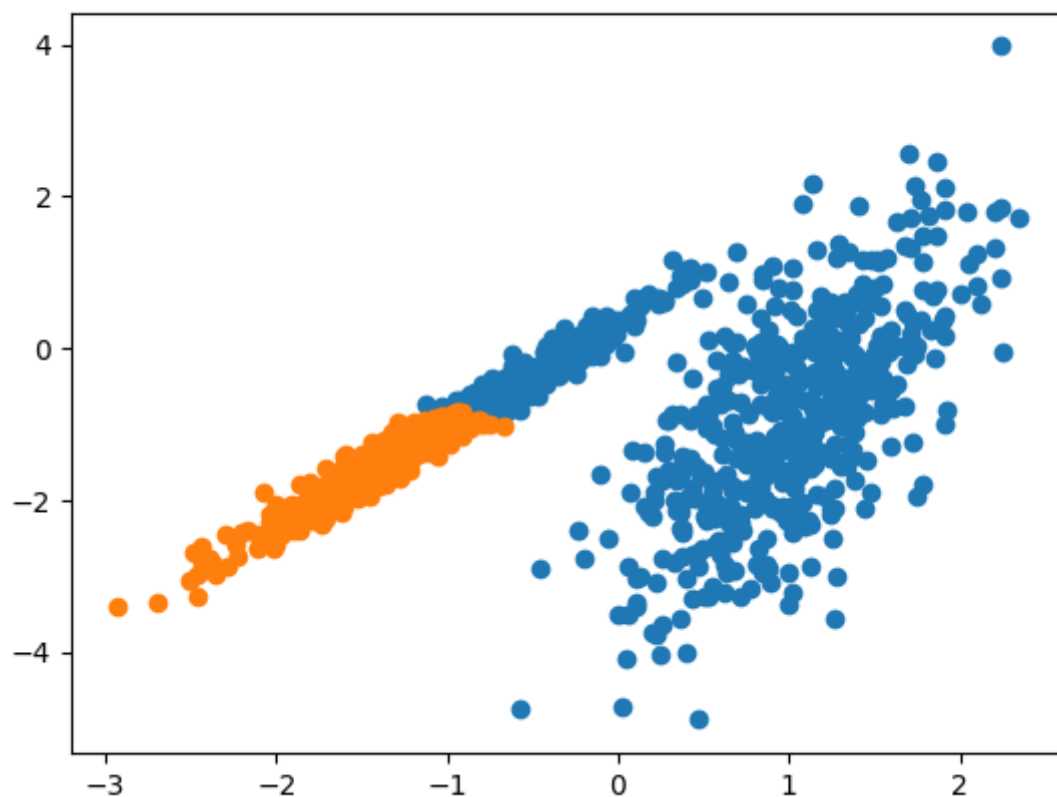
```
# na którym wyświetlane są próbki z cechami z pierwszej i drugiej kolumny danych X, które  
są przypisane do obecnie badanego klastra.
```

```
plt.scatter(X[row_ix, 0], X[row_ix, 1])
```

```
# wyświetlenie wykresu
```

```
plt.show()
```

ponizej wykres:

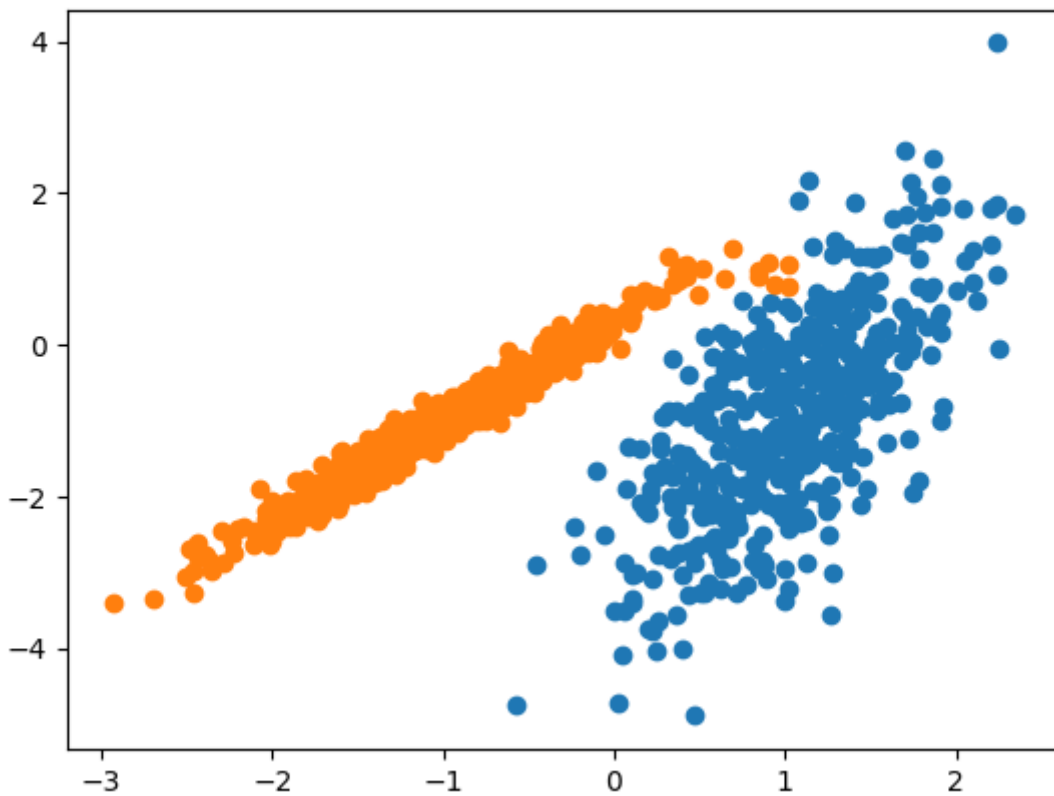


klasteryzacja , model Birch (python) kod wykorzystuje algorytm Birch do grupowania danych, a następnie przedstawia wyniki na wykresie punktowym

```
# Importuje funkcję unique z biblioteki NumPy, która zostanie użyta do znalezienia
unikalnych wartości w wektorze.
# Importuje funkcję where z biblioteki NumPy, która zostanie użyta do znalezienia indeksów,
które spełniają określone warunki.
# importuje funkcję make_classification z biblioteki Scikit-Learn, która zostanie użyta do
wygenerowania zestawu danych do eksperymentów z grupowaniem.
# Importuje klasę Birch z biblioteki Scikit-Learn, która reprezentuje algorytm grupowania
Birch
# Importuje moduł pyplot z biblioteki Matplotlib, który zostanie użyty do rysowania
wykresów.
from numpy import unique
from numpy import where
from sklearn.datasets import make_classification
from sklearn.cluster import Birch
from matplotlib import pyplot
# Generowanie zestawu danych z 2 cechami i 1000 próbkami za pomocą funkcji
make_classification. Parametry funkcji make_classification:
# - n_samples: Liczba próbek (1000).
# - n_features: Liczba cech (2).
# - n_informative: Liczba informatywnych cech (2) - cechy te wpływają na separację klas.
# - n_redundant: Liczba nadmiarowych cech (0) - cechy te są losowo wygenerowane i nie
wpływają na separację klas.
# - n_clusters_per_class: Liczba klastrów na klasę (1) - oznacza, że każda klasa jest
jednym skupiskiem danych.
# - random_state: Ziarno generatora liczb losowych (ustawione na 4 dla powtarzalności
eksperymentu).
X, _ = make_classification(n_samples=1000, n_features=2, n_informative=2, n_redundant=0,
n_clusters_per_class=1, random_state=4)
# inicjalizuje model Birch z określonymi parametrami. Parametr threshold kontroluje
wartość progową używaną przez algorytm Birch.
# Jest to próg odległości, który decyduje, ile próbek zostanie przydzielonych do węzła w
drzewie grupowania. Wartość 0.01 oznacza, że próg odległości
# między próbkami musi być mniejszy niż 0.01, aby były one przydzielane do tego samego
węzła. Wyższa wartość progowa spowoduje, że więcej próbek zostanie
# przydzielonych do jednego węzła, co może prowadzić do większych klastrów. clusters=2:
Parametr n_clusters określa, ile klastrów model ma znaleźć
# podczas grupowania. W tym przypadku ustawienie n_clusters na 2 oznacza, że model ma za
zadanie znaleźć 2 klastry w danych. Podsumowując,
# linia ta inicjalizuje model Birch z wartością progową wynoszącą 0.01 i zadaniem
znalezienia 2 klastrów w danych
model = Birch(threshold=0.01, n_clusters=2)
# Dopasowuje model do danych X. Algorytm Birch buduje drzewo grupowania na podstawie
danych.
model.fit(X)
# Przewiduje przynależność każdej próbki do klastra za pomocą wytrenowanego modelu i
zapisuje wyniki w yhat.
yhat = model.predict(X)
# Znajduje unikalne klastry w wynikach przewidywania yhat
clusters = unique(yhat)
```

```
# Rozpoczyna pętlę po unikalnych klastrach
for cluster in clusters:
    # Dla każdego klastra znajduje indeksy próbek, które należą do tego klastra.
    row_ix = where(yhat == cluster)
    # tworzy wykres punktowy, na którym wyświetla próbki z danego klastra, używając pierwszej
    i drugiej cechy z zestawu danych X.
    pyplot.scatter(X[row_ix, 0], X[row_ix, 1])
# wyświetla wykres
pyplot.show()
```

wykres:

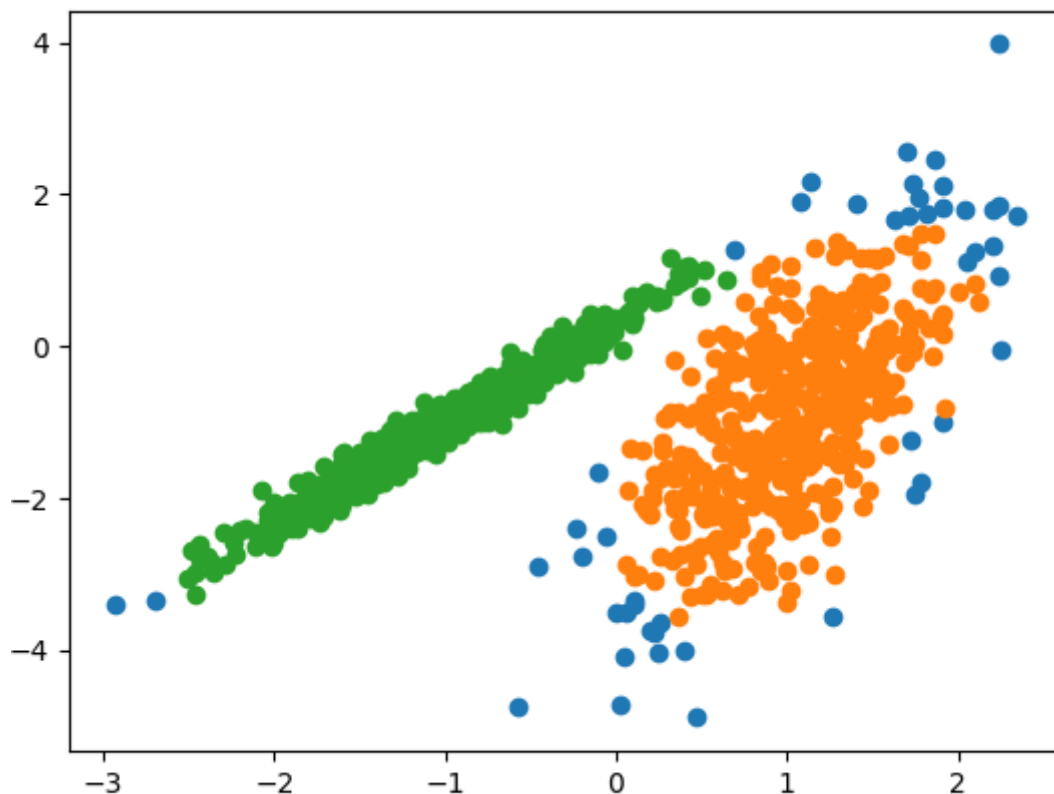


przykład zastosowania algorytmu klastrowania DBSCAN (Density-Based Spatial Clustering of Applications with Noise) do analizy danych. DBSCAN jest algorytmem klastrowania, który grupuje punkty danych na podstawie gęstości punktów w przestrzeni, identyfikując jednocześnie punkty odstające (szumy).

```
# Importuje funkcję unique z biblioteki NumPy, która zostanie użyta do znalezienia
unikalnych wartości w wektorze.
# Importuje funkcję where z biblioteki NumPy, która zostanie użyta do znalezienia indeksów,
które spełniają określone warunki.
# importuje funkcję make_classification z biblioteki Scikit-Learn, która zostanie użyta do
wygenerowania zestawu danych do eksperymentów z grupowaniem.
# Importuje klasę DBSCAN z biblioteki Scikit-Learn, która reprezentuje algorytm
klastrowania DBSCAN.
# Importuje moduł pyplot z biblioteki Matplotlib, który zostanie użyty do rysowania
wykresów.
from numpy import unique
from numpy import where
from sklearn.datasets import make_classification
from sklearn.cluster import DBSCAN
from matplotlib import pyplot
# Generowanie zestawu danych z 2 cechami i 1000 próbkami za pomocą funkcji
make_classification. Parametry funkcji make_classification:
# - n_samples: Liczba próbek (1000).
# - n_features: Liczba cech (2).
# - n_informative: Liczba informatywnych cech (2) - cechy te wpływają na separację klas.
# - n_redundant: Liczba nadmiarowych cech (0) - cechy te są losowo wygenerowane i nie
wpływają na separację klas.
# - n_clusters_per_class: Liczba klastrów na klasę (1) - oznacza, że każda klasa jest
jednym skupiskiem danych.
# - random_state: Ziarno generatora liczb losowych (ustawione na 4 dla powtarzalności
eksperymentu).
X, _ = make_classification(n_samples=1000, n_features=2, n_informative=2, n_redundant=0,
n_clusters_per_class=1, random_state=4)
# inicjalizowany jest model klastrowania DBSCAN (Density-Based Spatial Clustering of
Applications with Noise) z określonymi parametrami: eps to
# parametr "epsilon" ( $\epsilon$ ), który określa promień sąsiedztwa. Dla każdego punktu w
przestrzeni danych algorytm DBSCAN sprawdza, ile punktów znajduje się
# w odległości mniejszej lub równej eps. Wszystkie punkty, które mieszczą się w tym
promieniu sąsiedztwa, uważane są za sąsiednie i mogą być częścią
# tego samego klastra. min_samples to minimalna liczba punktów wymagana w sąsiedztwie danego
punktu, aby ten punkt został uznany za punkt centralny klastra.
# Jeśli w odległości eps od danego punktu znajduje się co najmniej min_samples innych
punktów, to ten punkt jest uznawany za punkt centralny klastra.
# W przeciwnym razie jest on uznawany za punkt odstający. Podsumowując, parametr eps
kontroluje, jak daleko algorytm DBSCAN będzie szukał sąsiednich
# punktów, podczas gdy min_samples kontroluje minimalną liczbę punktów w sąsiedztwie
wymaganą do utworzenia klastra. Te dwa parametry wpływają na to,
```

```
# jak algorytm identyfikuje klastry i punkty odstające w danych
model = DBSCAN(eps=0.30, min_samples=9)
# Dopasowuje model DBSCAN do danych X i przewiduje przynależność każdego punktu do klastra.
# Wynik jest przechowywany w zmiennej yhat.
yhat = model.fit_predict(X)
# Znajduje unikalne klastry przypisane przez model, używając funkcji unique.
clusters = unique(yhat)
# wykonuje pętlę po wszystkich znalezionych klastrach:
for cluster in clusters:
    # Dla każdego klastra znajduje indeksy punktów, które zostały przypisane do tego klastra.
    row_ix = where(yhat == cluster)
    # tworzy wykres punktowy, na którym wyświetla próbki z danego klastra, używając pierwszej
    # i drugiej cechy z zestawu danych X.
    pyplot.scatter(X[row_ix, 0], X[row_ix, 1])
# wyświetla wykres
pyplot.show()
```

wykres:

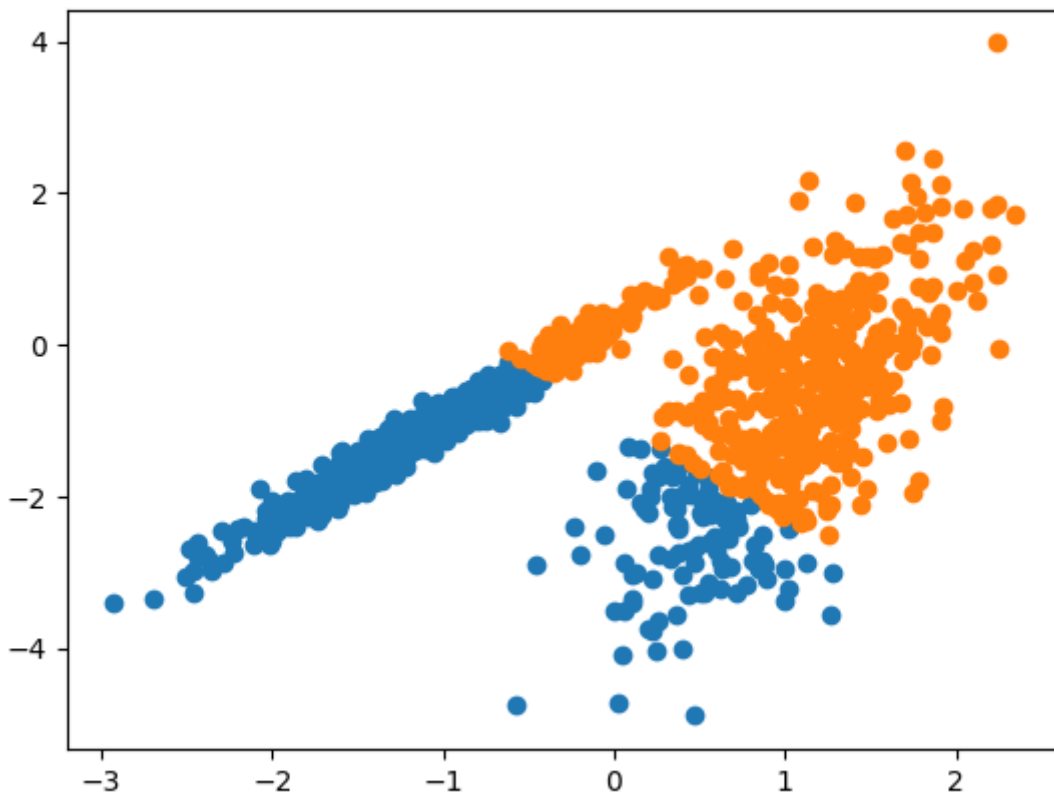


kod wykorzystuje algorytm k-means do podziału danych na klastry i wizualizuje te klastry na wykresie punktowym w zależności od przypisania klastra do punktów danych. K-means jest algorytmem klastrowania, który próbuje podzielić dane na grupy w oparciu o odległości między punktami danych i centrami klastrów.

```
# Importuje funkcję unique z biblioteki NumPy, która zostanie użyta do znalezienia
unikalnych wartości w wektorze.
# Importuje funkcję where z biblioteki NumPy, która zostanie użyta do znalezienia indeksów,
które spełniają określone warunki.
# importuje funkcję make_classification z biblioteki Scikit-Learn, która zostanie użyta do
wygenerowania zestawu danych do eksperymentów z grupowaniem.
# Importuje klasę KMeans z biblioteki Scikit-Learn, która reprezentuje algorytm k-means
clustering.
# Importuje moduł pyplot z biblioteki Matplotlib, który zostanie użyty do rysowania
wykresów.
from numpy import unique
from numpy import where
from sklearn.datasets import make_classification
from sklearn.cluster import KMeans
from matplotlib import pyplot
# Generowanie zestawu danych z 2 cechami i 1000 próbkami za pomocą funkcji
make_classification. Parametry funkcji make_classification:
# - n_samples: Liczba próbek (1000).
# - n_features: Liczba cech (2).
# - n_informative: Liczba informatywnych cech (2) - cechy te wpływają na separację klas.
# - n_redundant: Liczba nadmiarowych cech (0) - cechy te są losowo wygenerowane i nie
wpływają na separację klas.
# - n_clusters_per_class: Liczba klastrów na klasę (1) - oznacza, że każda klasa jest
jednym skupiskiem danych.
# - random_state: Ziarno generatora liczb losowych (ustawione na 4 dla powtarzalności
eksperymentu).
X, _ = make_classification(n_samples=1000, n_features=2, n_informative=2, n_redundant=0,
n_clusters_per_class=1, random_state=4)
# model k-means clustering z określoną liczbą klastrów (n_clusters). W tym przypadku
ustawiona jest liczba klastrów na 2.
model = KMeans(n_clusters=2)
# Dopasowuje model k-means do danych X, co oznacza, że algorytm k-means będzie próbował
znaleźć optymalne centra klastrów na podstawie danych.
model.fit(X)
# Przypisuje każdemu przykładowi z danych X przynależność do klastra na podstawie
dopasowanego modelu. Każdy przykład zostaje przyporządkowany
# do najbliższego klastra na podstawie odległości od centrów klastrów.
yhat = model.predict(X)
# Znajduje unikalne klastry przypisane przez model, używając funkcji unique.
clusters = unique(yhat)
```

```
# wykonuje pętlę po wszystkich znalezionych klastrach:  
for cluster in clusters:  
    # Dla każdego klastra znajduje indeksy punktów, które zostały przypisane do tego klastra.  
    row_ix = where(yhat == cluster)  
    # tworzy wykres punktowy, na którym wyświetla próbki z danego klastra, używając pierwszej  
    i drugiej cechy z zestawu danych X.  
    pyplot.scatter(X[row_ix, 0], X[row_ix, 1])  
# wyświetla wykres  
pyplot.show()
```

wykres:

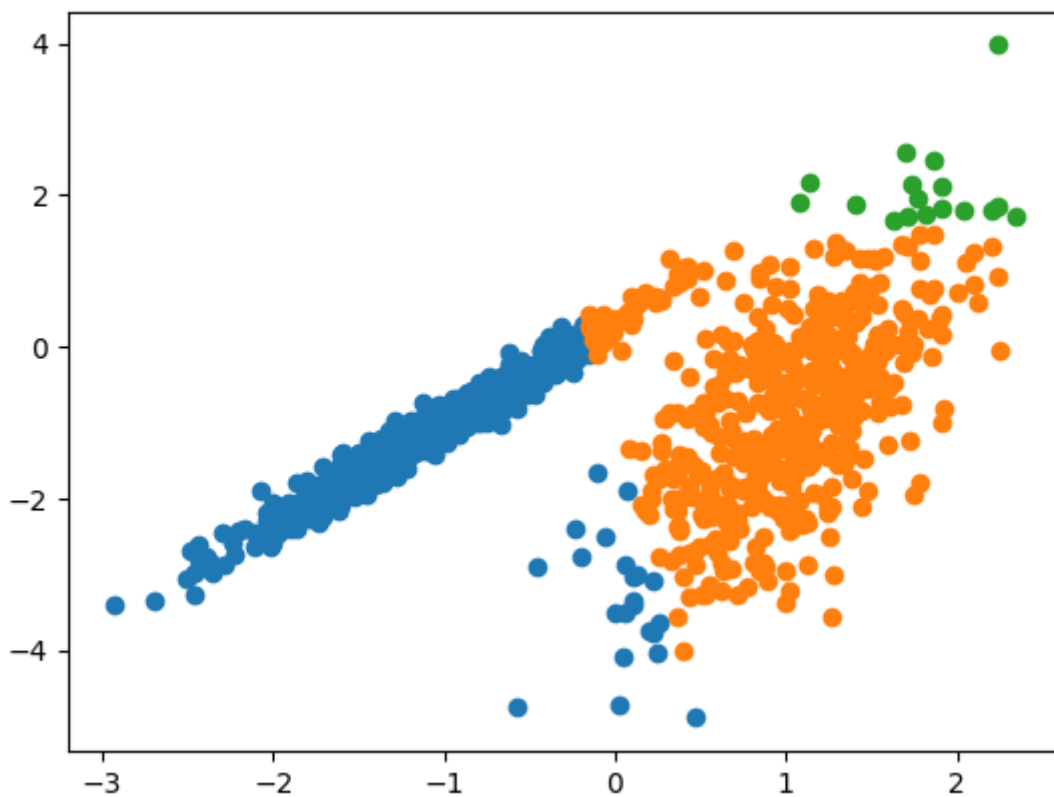


przykład zastosowania algorytmu klastrowania Mean Shift do analizy danych. Algorytm Mean Shift jest techniką klastrowania, która nie wymaga wcześniejszej wiedzy o liczbie klastrów i próbuje samodzielnie znaleźć optymalne centra klastrów na podstawie danych.

```
# Importuje funkcję unique z biblioteki NumPy, która zostanie użyta do znalezienia
unikalnych wartości w wektorze.
# Importuje funkcję where z biblioteki NumPy, która zostanie użyta do znalezienia indeksów,
które spełniają określone warunki.
# importuje funkcję make_classification z biblioteki Scikit-Learn, która zostanie użyta do
wygenerowania zestawu danych do eksperymentów z grupowaniem.
# Importuje klasę MeanShift z biblioteki Scikit-Learn, która reprezentuje algorytm Mean
Shift Clustering.
# Importuje moduł pyplot z biblioteki Matplotlib, który zostanie użyty do rysowania
wykresów.
from numpy import unique
from numpy import where
from sklearn.datasets import make_classification
from sklearn.cluster import MeanShift
from matplotlib import pyplot
# Generowanie zestawu danych z 2 cechami i 1000 próbkami za pomocą funkcji
make_classification. Parametry funkcji make_classification:
# - n_samples: Liczba próbek (1000).
# - n_features: Liczba cech (2).
# - n_informative: Liczba informatywnych cech (2) - cechy te wpływają na separację klas.
# - n_redundant: Liczba nadmiarowych cech (0) - cechy te są losowo wygenerowane i nie
wpływają na separację klas.
# - n_clusters_per_class: Liczba klastrów na klasę (1) - oznacza, że każda klasa jest
jednym skupiskiem danych.
# - random_state: Ziarno generatora liczb losowych (ustawione na 4 dla powtarzalności
eksperymentu).
X, _ = make_classification(n_samples=1000, n_features=2, n_informative=2, n_redundant=0,
n_clusters_per_class=1, random_state=4)
# inicjalizuje model Mean Shift Clustering bez określania dodatkowych parametrów. Algorytm
Mean Shift sam próbuje wyznaczyć optymalne centra klastrów
# na podstawie danych.
model = MeanShift()
# Dopasowuje model Mean Shift do danych X i przewiduje przynależność każdego punktu do
klastra. Wynik jest przechowywany w zmiennej yhat.
yhat = model.fit_predict(X)
#Znajduje unikalne klastry przypisane przez model, używając funkcji unique.
clusters = unique(yhat)
# wykonuje pętlę po wszystkich znalezionych klastrach:
for cluster in clusters:
    # Dla każdego klastra znajduje indeksy punktów, które zostały przypisane do tego klastra.
```

```
row_ix = where(yhat == cluster)
# tworzy wykres punktowy, na którym wyświetla próbki z danego klastra, używając pierwszej
i drugiej cechy z zestawu danych X.
pyplot.scatter(X[row_ix, 0], X[row_ix, 1])
# wyświetla wykres
pyplot.show()
```

wykres:

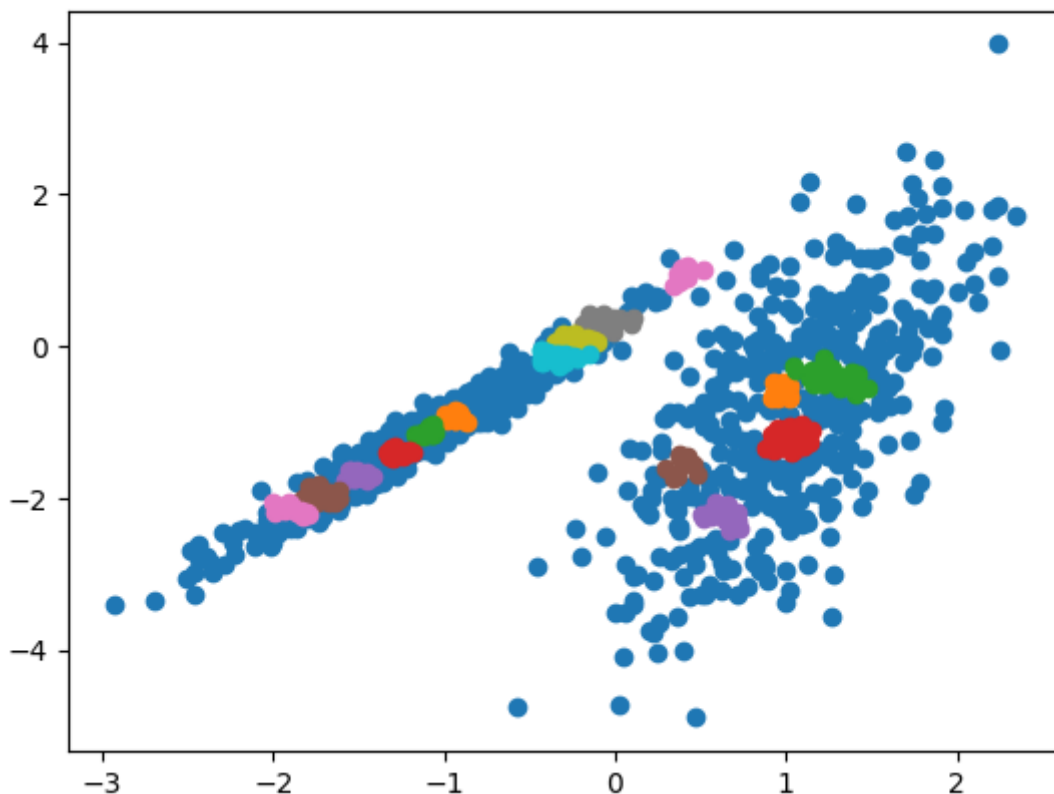


Algorytm OPTICS jest używany do klastrowania danych, a jego celem jest identyfikacja klastrów różnej gęstości i kształtu. Ten kod demonstruje, jak używać algorytmu OPTICS do analizy danych i wizualizacji wyników klastrowania na wykresie punktowym.

```
# Importuje funkcję unique z biblioteki NumPy, która zostanie użyta do znalezienia
unikalnych wartości w wektorze.
# Importuje funkcję where z biblioteki NumPy, która zostanie użyta do znalezienia indeksów,
które spełniają określone warunki.
# importuje funkcję make_classification z biblioteki Scikit-Learn, która zostanie użyta do
wygenerowania zestawu danych do eksperymentów z grupowaniem.
# Importuje klasę OPTICS z biblioteki Scikit-Learn, która reprezentuje algorytm OPTICS do
klastrowania danych.
# Importuje moduł pyplot z biblioteki Matplotlib, który zostanie użyty do rysowania
wykresów.
from numpy import unique
from numpy import where
from sklearn.datasets import make_classification
from sklearn.cluster import OPTICS
from matplotlib import pyplot
# Generowanie zestawu danych z 2 cechami i 1000 próbkami za pomocą funkcji
make_classification. Parametry funkcji make_classification:
# - n_samples: Liczba próbek (1000).
# - n_features: Liczba cech (2).
# - n_informative: Liczba informatywnych cech (2) - cechy te wpływają na separację klas.
# - n_redundant: Liczba nadmiarowych cech (0) - cechy te są losowo wygenerowane i nie
wpływają na separację klas.
# - n_clusters_per_class: Liczba klastrów na klasę (1) - oznacza, że każda klasa jest
jednym skupiskiem danych.
# - random_state: Ziarno generatora liczb losowych (ustawione na 4 dla powtarzalności
eksperymentu).
X, _ = make_classification(n_samples=1000, n_features=2, n_informative=2, n_redundant=0,
n_clusters_per_class=1, random_state=4)
# W tym wierszu kodu inicjalizowany jest model OPTICS z określonymi parametrami: eps to
parametr "epsilon" ( $\epsilon$ ), który określa promień sąsiedztwa w
# algorytmie OPTICS. Parametr ten definiuje maksymalną odległość, na jaką algorytm będzie
szukał sąsiednich punktów dla każdego punktu w zbiorze danych.
# Wartość eps wpływa na to, jak daleko punkty muszą być od siebie, aby były uważane za
sąsiednie. min_samples to minimalna liczba punktów w sąsiedztwie
# wymagana do uznania punktu za punkt rdzeniowy (core point) w algorytmie OPTICS. Punkt
rdzeniowy to punkt, który ma przynajmniej min_samples
# innych punktów w swoim eps-odległości sąsiedztwa. Punkty rdzeniowe stanowią centralną
część klastra. Algorytm OPTICS wykorzystuje te parametry
```

```
# do analizy struktury klastrow w danych. Działa poprzez budowanie drzewa struktury
klastrow, które ukazuje hierarchiczne relacje między klastrami
# różnej gęstości. Parametry eps i min_samples wpływają na to, jakie klastry będą wykrywane
i jakie punkty zostaną uznane za punkty rdzeniowe.
model = OPTICS(eps=0.8, min_samples=10)
# Dopasowuje model OPTICS do danych X i przewiduje przynależność każdego punktu do
klastra. Wynik jest przechowywany w zmiennej yhat.
yhat = model.fit_predict(X)
#Znajduje unikalne klastry przypisane przez model, używając funkcji unique.
clusters = unique(yhat)
# wykonuje pętlę po wszystkich znalezionych klastrach:
for cluster in clusters:
    # Dla każdego klastra znajduje indeksy punktów, które zostały przypisane do tego klastra.
    row_ix = where(yhat == cluster)
    # tworzy wykres punktowy, na którym wyświetla próbki z danego klastra, używając pierwszej
i drugiej cechy z zestawu danych X.
    pyplot.scatter(X[row_ix, 0], X[row_ix, 1])
# wyświetla wykres
pyplot.show()
```

wykres:



kod jest przykładem zastosowania algorytmu Gaussian Mixture Model (GMM) do analizy danych i klastrowania. GMM jest modelem probabilistycznym, który zakłada, że dane pochodzą z mieszaniny wielu rozkładów Gaussa

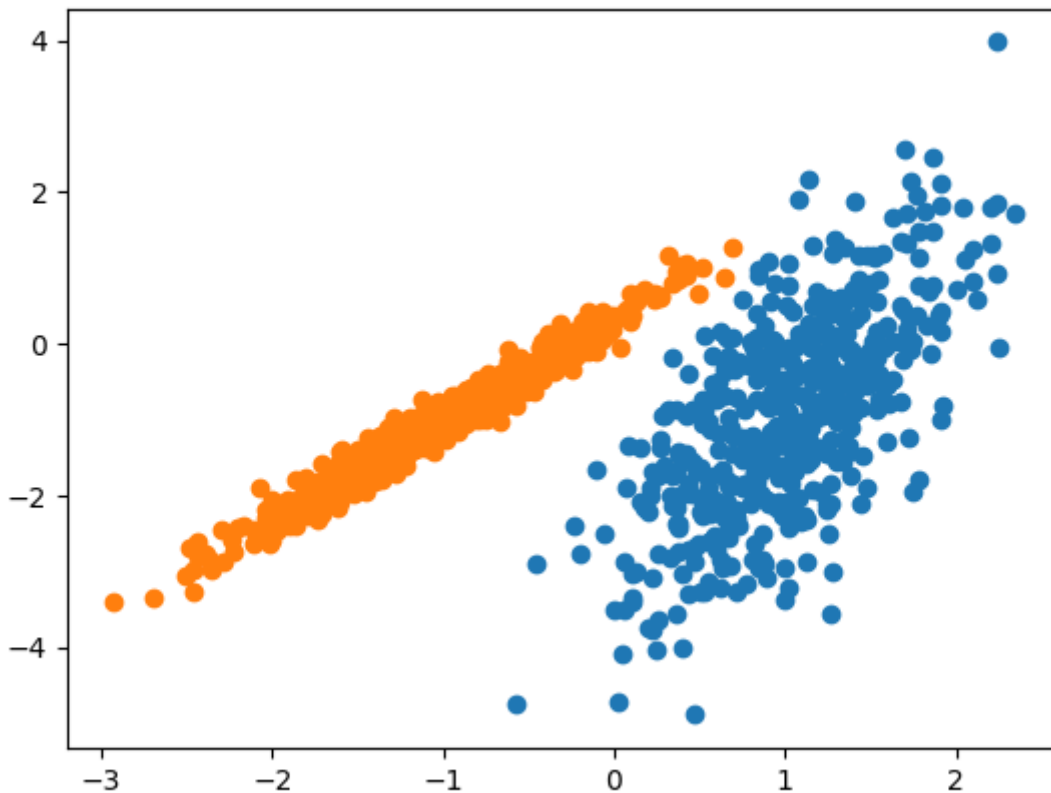
```
# Importuje funkcję unique z biblioteki NumPy, która zostanie użyta do znalezienia
unikalnych wartości w wektorze.
# Importuje funkcję where z biblioteki NumPy, która zostanie użyta do znalezienia indeksów,
które spełniają określone warunki.
# importuje funkcję make_classification z biblioteki Scikit-Learn, która zostanie użyta do
wygenerowania zestawu danych do eksperymentów z grupowaniem.
# Importuje klasę GaussianMixture z biblioteki Scikit-Learn, która reprezentuje algorytm
klastrowania Gaussian Mixture Model (GMM).
# Importuje moduł pyplot z biblioteki Matplotlib, który zostanie użyty do rysowania
wykresów.
from numpy import unique
from numpy import where
from sklearn.datasets import make_classification
from sklearn.mixture import GaussianMixture
from matplotlib import pyplot
# Generowanie zestawu danych z 2 cechami i 1000 próbkami za pomocą funkcji
make_classification. Parametry funkcji make_classification:
# - n_samples: Liczba próbek (1000).
# - n_features: Liczba cech (2).
# - n_informative: Liczba informatywnych cech (2) - cechy te wpływają na separację klas.
# - n_redundant: Liczba nadmiarowych cech (0) - cechy te są losowo wygenerowane i nie
wpływają na separację klas.
# - n_clusters_per_class: Liczba klastrów na klasę (1) - oznacza, że każda klasa jest
jednym skupiskiem danych.
# - random state: Ziarno generatora liczb losowych (ustawione na 4 dla powtarzalności
eksperymentu).
X, _ = make_classification(n_samples=1000, n_features=2, n_informative=2, n_redundant=0,
n_clusters_per_class=1, random_state=4)
# Inicjalizuje model Gaussian Mixture Model (GMM) z dwoma komponentami (klastrami) za
pomocą parametru n_components.
model = GaussianMixture(n_components=2)
# Dopasowuje model GMM do danych, aby oszacować parametry rozkładów Gaussa, które najlepiej
pasują do danych.
model.fit(X)
```

```

# Przypisuje każdemu przykładowi z danych przynależność do jednego z klastrów na podstawie
oszacowanych rozkładów.
yhat = model.predict(X)
#Znajduje unikalne klastry przypisane przez model, używając funkcji unique.
clusters = unique(yhat)
# wykonuje pętlę po wszystkich znalezionych klastrach:
for cluster in clusters:
    # Dla każdego klastra znajduje indeksy punktów, które zostały przypisane do tego klastra.
    row_ix = where(yhat == cluster)
    # tworzy wykres punktowy, na którym wyświetla próbki z danego klastra, używając pierwszej
i drugiej cechy z zestawu danych X.
    pyplot.scatter(X[row_ix, 0], X[row_ix, 1])
# wyświetla wykres
pyplot.show()

```

wykres:



Przykład praktycznego użycia klasteryzacji w logistyce

np w przypadku: przykład grupowania produktów podobnych pod względem popytu, używając algorytmu DBSCAN.

```
#Importuje bibliotekę NumPy, popularny pakiet do pracy z obliczeniami numerycznymi w
Pythonie, i nadaje jej alias np. NumPy to popularny pakiet
# do obliczeń numerycznych w języku Python, który zawiera wiele przydatnych funkcji i
narzędzi do pracy z macierzami, tablicami oraz innymi strukturami
# danych numerycznych.
#Importuje bibliotekę Matplotlib, która jest używana do tworzenia wykresów i wizualizacji
danych, i nadaje jej alias plt
# Importuje algorytm DBSCAN (Density-Based Spatial Clustering of Applications with Noise) z
biblioteki Scikit-Learn, który jest używany
# do klastrowania danych na podstawie gęstości przestrzennej.
# Importuje funkcje unique i where z biblioteki NumPy. unique jest używane do pobrania
unikalnych wartości, a where służy do znalezienia indeksów,
# gdzie określony warunek jest spełniony.
# Importuje bibliotekę Pandas, która jest używana do manipulacji danymi i tworzenia
DataFrame, i nadaje jej alias pd.DataFrame to struktura danych,
# która jest często używana do przechowywania i analizy danych tabelarycznych, takich jak
dane z arkusza kalkulacyjnego lub bazy danych.
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import DBSCAN
from numpy import unique, where
import pandas as pd

# Przykładowe dane: popyt i cena produktów (40 punktów danych).Definiuje zestaw danych, w
którym każdy wiersz to dane dotyczące produktu,
# gdzie pierwsza kolumna reprezentuje popyt, a druga kolumna cenę.
data = np.array([
    [5, 10],
    [8, 5],
    [3, 2],
    [9, 8],
    [2, 1],
    [7, 6],
```

```
[4, 3],
[6, 9],
[1, 4],
[3, 7],
[2, 9],
[8, 3],
[7, 2],
[5, 4],
[4, 6],
[9, 7],
[8, 2],
[6, 4],
[2, 5],
[3, 8],
[7, 9],
[1, 2],
[4, 1],
[5, 7],
[9, 5],
[1, 8],
[8, 7],
[6, 3],
[2, 4],
[7, 1],
[3, 9],
[5, 3],
[4, 8],
[1, 6],
[9, 2],
[6, 1],
[8, 6],
[2, 7],
[4, 5]
])
```

```
# Tworzy model klastrowania DBSCAN z określonymi parametrami. eps to promień sąsiedztwa, a
min samples to minimalna liczba punktów w sąsiedztwie,
# aby punkt został uznany za część klastra.
model = DBSCAN(eps=1, min_samples=4)
```

```
# Dopasowanie modelu i przypisanie etykiet klastrów
labels = model.fit_predict(data)
```

```
# Tworzenie DataFrame z produktami i ich przypisanymi klastrami.product_names: To jest
zmienna, która przechowuje listę nazw produktów.
#Ta lista zostanie użyta później do utworzenia kolumny "Produkt" w DataFrame.f'Produkt
{i+1}' for i in range(len(data))]:
#To jest wyrażenie list comprehension. Pozwala ono na utworzenie listy, wykonując pewne
operacje na każdym elemencie z zakresu określonego przez
# for i in range(len(data)).
#List comprehension to technika używana w języku Python do tworzenia nowych list na
podstawie istniejących list lub innych iterowalnych obiektów.
# Jest to wygodny i zwięzły sposób tworzenia listy, który pozwala na przekształcanie i
filtrowanie danych w jednej linii kodu.
# for i in range(len(data)): To jest pętla for, która przechodzi przez zakres liczb od 0 do
len(data)Wyrażenie len(data) zwraca liczbę
# wierszy tej tablicy, co jest równoznaczne z liczbą produktów w zestawie danych
```

```

# f'Produkt {i+1}': To jest formatowany napis, który jest tworzony dla każdego i w
zakresie. i+1 jest używane do utworzenia numeracji produktów,
# zaczynając od 1 (ponieważ indeksowanie w Pythonie zaczyna się od 0)
#Całe wyrażenie generuje listę nazw produktów w formacie "Produkt 1", "Produkt 2", "Produkt
3", itd., w zależności od liczby produktów w tablicy data.
#Ostatecznie, lista product_names zawiera nazwy produktów, które zostaną użyte do
utworzenia kolumny "Produkt" w DataFrame cluster_df.
#każda nazwa produktu odpowiada jednemu produktowi w zestawie danych.
product_names = [f'Produkt {i+1}' for i in range(len(data))]
# cluster_df: To jest DataFrame utworzony przy użyciu biblioteki Pandas. DataFrame ten
zawiera dwie kolumny: Kolumna "Produkt":
# Zawiera nazwy produktów z listy product_names. Każdy wiersz odpowiada jednemu produktowi,
a nazwy produktów są przypisane do tej kolumny.
# Kolumna "Klaster": Zawiera etykiety klastrów przypisane do produktów zmienną labels.
Każda etykieta klastra jest przypisana do produktu na
#podstawie wyników klastrowania za pomocą algorytmu DBSCAN.
#W efekcie końcowym, DataFrame cluster_df zawiera informacje o produktach i przypisanych im
klastrach, co pozwala na dalszą analizę danych w formie tabelarycznej.
cluster_df = pd.DataFrame({'Produkt': product_names, 'Klaster': labels})

# Sortuje dane w DataFrame cluster_df według klastrów, tworząc nowy DataFrame
cluster_df_sorted.
cluster_df_sorted = cluster_df.sort_values(by=['Klaster'])

# Wydruk posortowanej tabeli
print(cluster_df_sorted)

# Wizualizacja wyników
plt.figure(figsize=(10, 6))

# Wykres produktów w klastrach
for cluster in np.unique(labels):
    # Pobranie indeksów próbek w danym klastrze
    row_ix = where(labels == cluster)
    # Wyodrębnienie próbek w danym klastrze
    cluster_data = data[row_ix]
    # Oznaczenie kolorami cen i popytu
    plt.scatter(cluster_data[:, 0], cluster_data[:, 1], label=f'Cluster {cluster + 1}')

plt.xlabel('Popyt')
plt.ylabel('Cena')
plt.legend()
plt.title('Grupowanie produktów podobnych pod względem popytu i ceny')
plt.show()

```

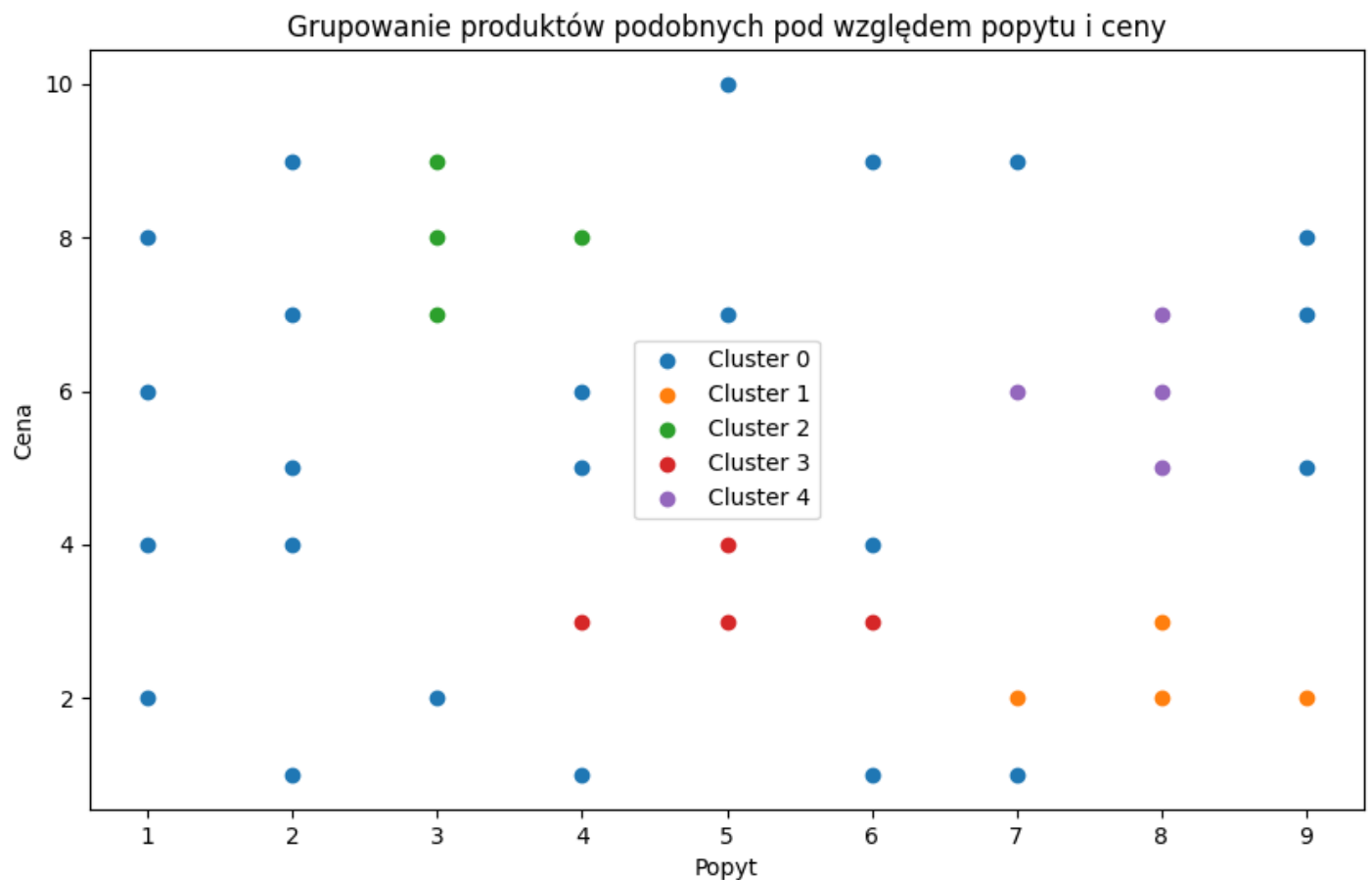
tabela danych:

Produkt Klaster

0	Produkt 1	-1
35	Produkt 36	-1
33	Produkt 34	-1
29	Produkt 30	-1
28	Produkt 29	-1

25	Produkt 26	-1
24	Produkt 25	-1
23	Produkt 24	-1
22	Produkt 23	-1
21	Produkt 22	-1
20	Produkt 21	-1
37	Produkt 38	-1
18	Produkt 19	-1
17	Produkt 18	-1
15	Produkt 16	-1
38	Produkt 39	-1
10	Produkt 11	-1
2	Produkt 3	-1
8	Produkt 9	-1
7	Produkt 8	-1
3	Produkt 4	-1
14	Produkt 15	-1
4	Produkt 5	-1
16	Produkt 17	0
34	Produkt 35	0
12	Produkt 13	0
11	Produkt 12	0
32	Produkt 33	1
30	Produkt 31	1
19	Produkt 20	1
9	Produkt 10	1
31	Produkt 32	2
6	Produkt 7	2
13	Produkt 14	2
27	Produkt 28	2
5	Produkt 6	3
1	Produkt 2	3
36	Produkt 37	3
26	Produkt 27	3

wykres:



Kolejny przykład klastrowanie, grupowanie produktow na magazynie :

Optymalizacja układu regałów często opiera się na grupowaniu produktów podobnych pod względem rozmiarów, kształtów czy wymagań przechowywania. W tym przypadku algorytm K-means może być dobrym wyborem, ponieważ próbuje podzielić dane na klastry o zbliżonych średnich, co może pomóc w efektywnym umieszczeniu produktów w magazynie

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from numpy import unique, where
```

```
# Przykładowe dane: rozmiar i waga produktów (40 punktów danych)
data = np.array([
    [20, 10],
    [15, 8],
    [25, 12],
    [30, 14],
    [10, 6],
    [18, 9],
```

```
[22, 11],  
[28, 13],  
[12, 7],  
[24, 12],  
[14, 7],  
[26, 13],  
[19, 9],  
[13, 6],  
[27, 11],  
[17, 8],  
[11, 6],  
[21, 10],  
[29, 14],  
[16, 8],  
[23, 11],  
[31, 15],  
[15, 7],  
[14, 6],  
[28, 12],  
[20, 9],  
[24, 11],  
[11, 5],  
[22, 10],  
[19, 8],  
[26, 12],  
[16, 7],  
[27, 13],  
[12, 6],  
[18, 8],  
[23, 10],  
[17, 7],  
[25, 13],  
])
```

```
# Definiowanie modelu K-means z określoną liczbą klastrów (np. 3)  
num_clusters = 3  
kmeans = KMeans(n_clusters=num_clusters)
```

```
# Dopasowanie modelu  
kmeans.fit(data)
```

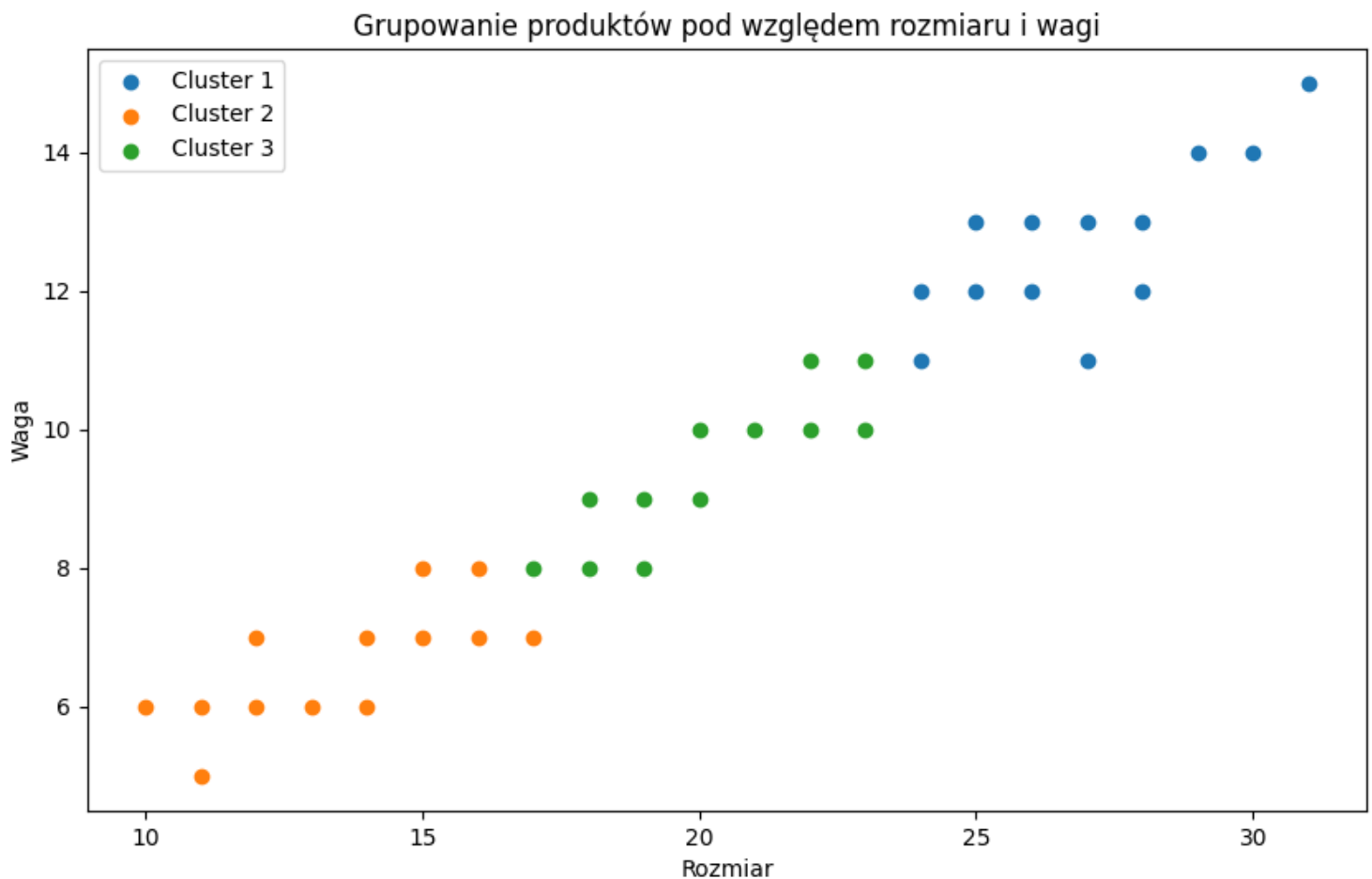
```
# Przypisanie klastrów do danych  
labels = kmeans.labels_
```

```
# Wizualizacja wyników  
plt.figure(figsize=(10, 6))
```

```
# Wykres produktów w klastrach z różnymi kolorami  
for cluster in np.unique(labels):  
    row_ix = where(labels == cluster)  
    cluster_data = data[row_ix]  
    plt.scatter(cluster_data[:, 0], cluster_data[:, 1], label=f'Cluster {cluster + 1}')
```

```
plt.xlabel('Rozmiar')  
plt.ylabel('Waga')  
plt.legend()  
plt.title('Grupowanie produktów pod względem rozmiaru i wagi')
```

```
plt.show()
```



kolejny przykład: Jeśli dane wykazują bardziej złożone wzorce i klastry o różnych kształtach, warto rozważyć użycie Gaussian Mixture Model (GMM), który jest elastyczny i potrafi modelować różne rozkłady w danych.

w tym wypadku sa 3 cechy

wysokosz, szerokosc, dlugosc produktu

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.mixture import GaussianMixture
from mpl_toolkits.mplot3d import Axes3D
import pandas as pd
```

```
# Przykładowe dane: szerokość, wysokość i długość produktów (40 punktów danych)
data = np.array([
    [20, 10, 30],
    [15, 8, 25],
    [25, 12, 35],
    [30, 14, 40],
```

```
[10, 6, 15],  
[18, 9, 28],  
[22, 11, 32],  
[28, 13, 38],  
[12, 7, 20],  
[24, 12, 33],  
1])
```

```
# Definiowanie modelu GMM z określoną liczbą komponentów (klastrów)
```

```
num_components = 3
```

```
gmm = GaussianMixture(n_components=num_components)
```

```
# Dopasowanie modelu
```

```
gmm.fit(data)
```

```
# Przypisanie klastrów do danych
```

```
labels = gmm.predict(data)
```

```
# Tworzenie DataFrame z produktami i ich przypisanymi klastrami
```

```
product_names = [f'Produkt {i+1}' for i in range(len(data))]
```

```
cluster_df = pd.DataFrame({'Produkt': product_names, 'Klaster': labels})
```

```
# Sortowanie danych po klastrach
```

```
cluster_df_sorted = cluster_df.sort_values(by=['Klaster'])
```

```
# Wydruk posortowanej tabeli
```

```
print(cluster_df_sorted)
```

```
# Wizualizacja wyników w trójwymiarowym wykresie
```

```
fig = plt.figure(figsize=(10, 6))
```

```
ax = fig.add_subplot(111, projection='3d')
```

```
# Wykres produktów w klastrach z różnymi kolorami
```

```
for cluster in np.unique(labels):
```

```
    cluster_data = data[labels == cluster]
```

```
    ax.scatter(cluster_data[:, 0], cluster_data[:, 1], cluster_data[:, 2], label=f'Cluster  
{cluster + 1}')
```

```
ax.set_xlabel('Szerokość')
```

```
ax.set_ylabel('Wysokość')
```

```
ax.set_zlabel('Długość')
```

```
plt.legend()
```

```
plt.title('Grupowanie produktów pod względem trzech cech za pomocą GMM (3D)')
```

```
plt.show()
```

wykres:

Grupowanie produktów pod względem wielu cech za pomocą GMM (3D)

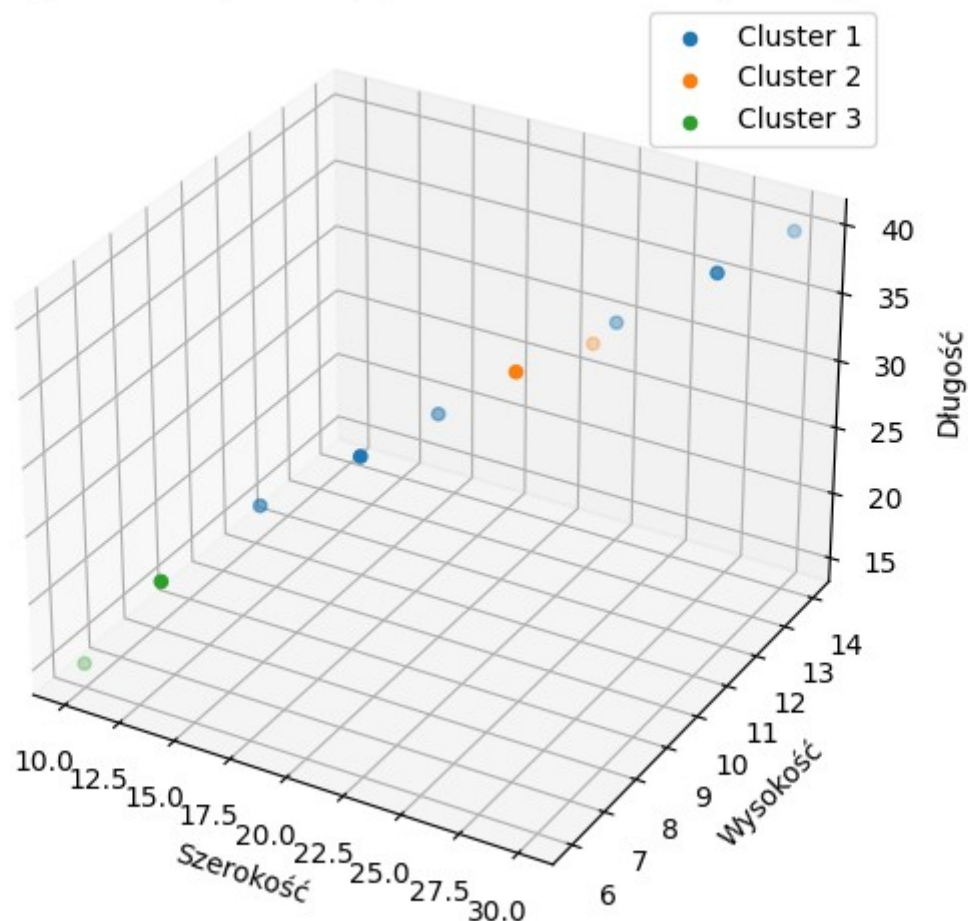


tabela danych:

Produkt Klaster

0	Produkt 1	0
1	Produkt 2	0
2	Produkt 3	0
3	Produkt 4	0
5	Produkt 6	0
7	Produkt 8	0
6	Produkt 7	1
9	Produkt 10	1
4	Produkt 5	2
8	Produkt 9	2

Załącznik 2. Nazwa załącznika 2

COS TAM TAM COS TAM