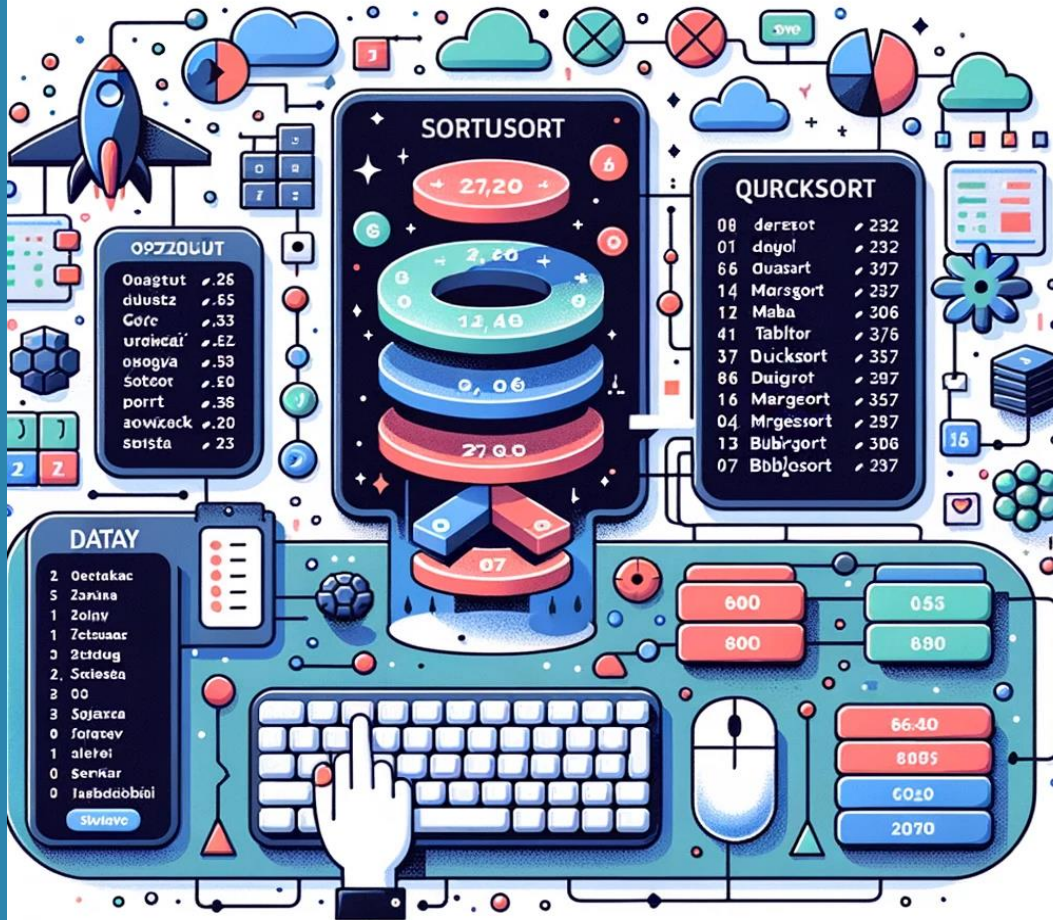




Przypadki użycia

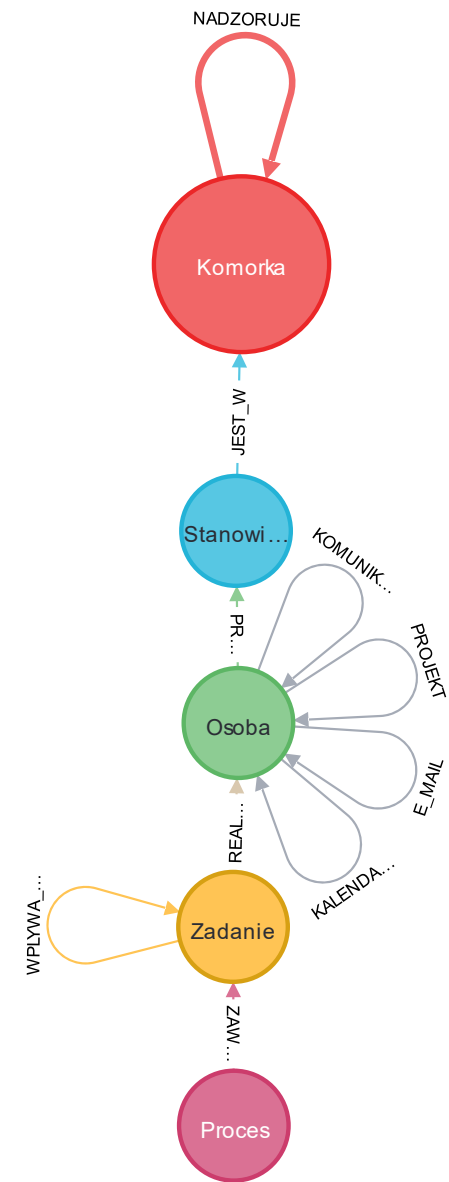
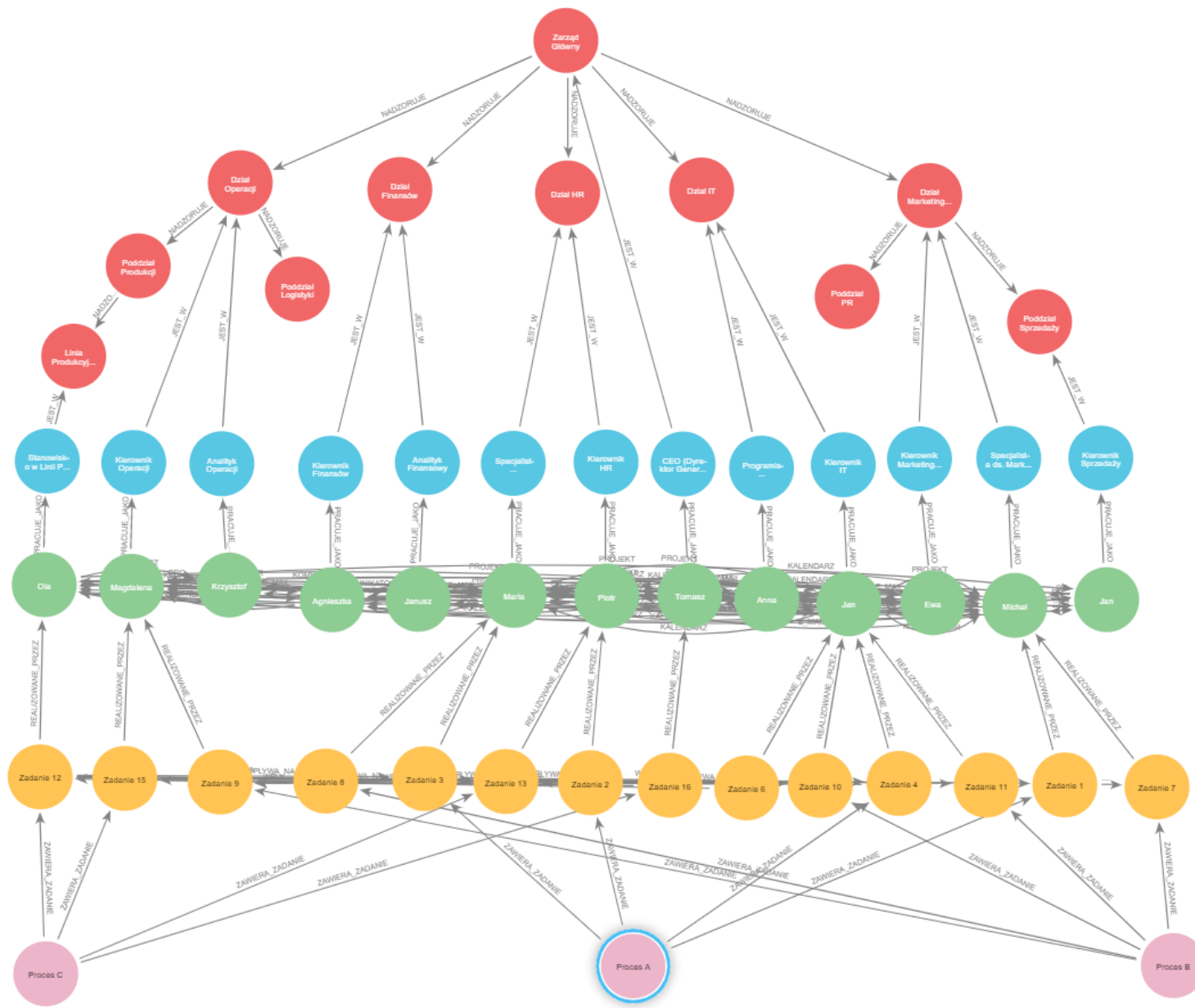
ANALIZA DANYCH
Z WYKORZYSTANIEM BAZY
GRAFOWEJ Neo4J
I ALGORYTMÓW GRAFOWYCH



Zagadnienia

- [Graf organizacji](#)
- [Mapowanie zależności zadań](#)
- [Detekcja liderów](#)

Heterogeniczny graf organizacji



Mapowanie zależności zadań

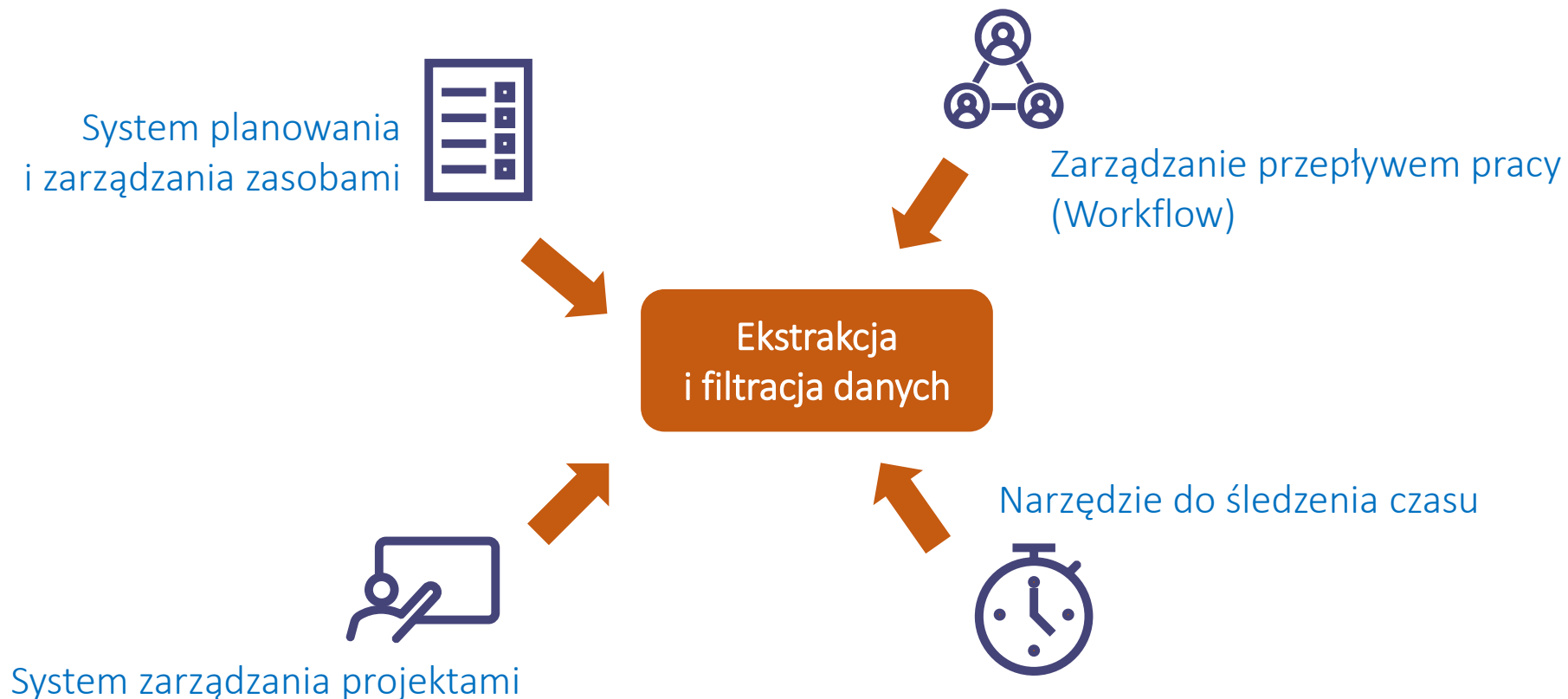
Wizualizacja i zrozumienie złożonych zależności między zadaniami, co jest nieocenione przy planowaniu projektów i alokacji zasobów.



Mapowanie zależności zadań

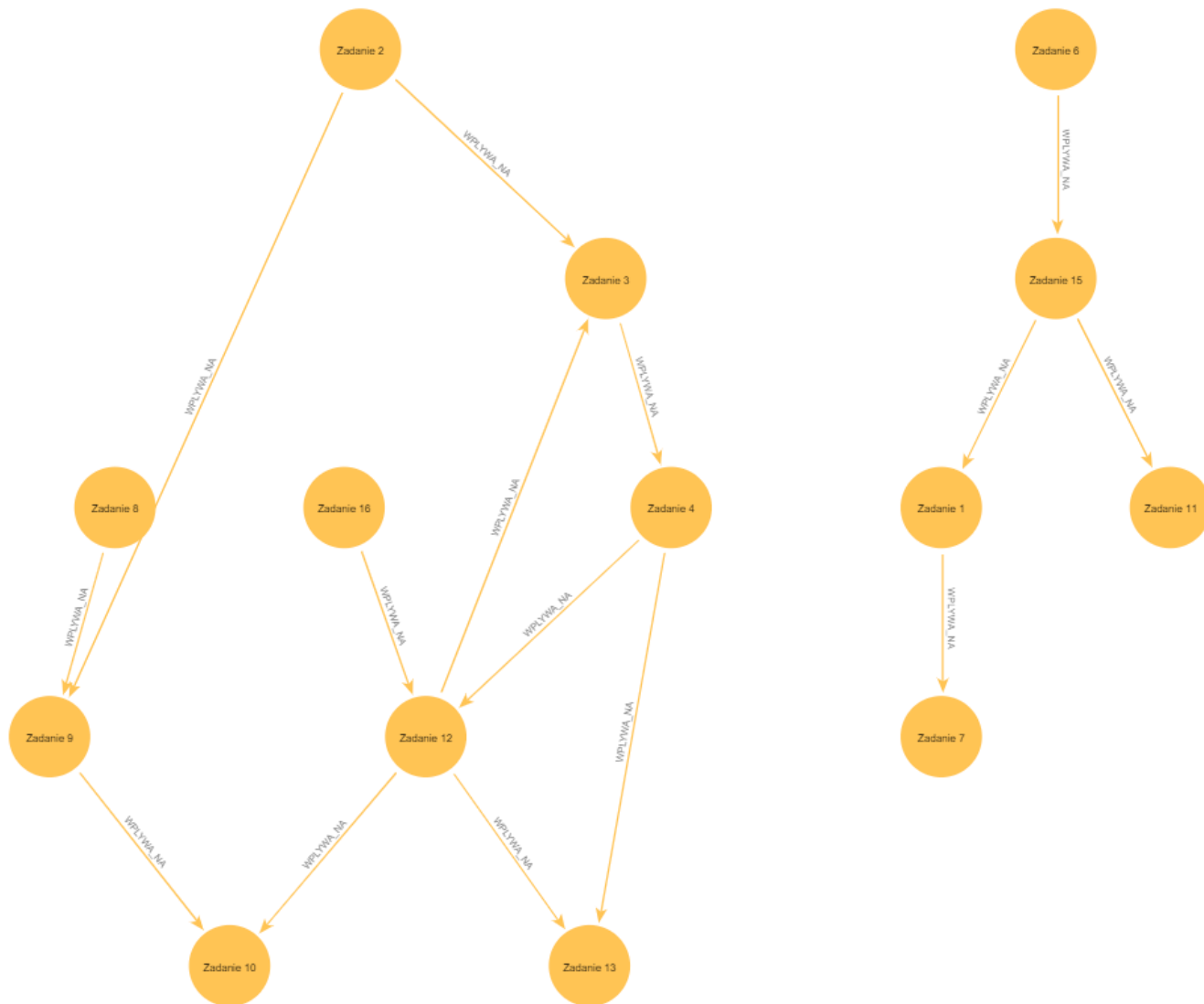
Eksploracja Danych

Zebranie danych o zadaniach realizowanych w organizacji w ramach procesów i projektów.



Tworzenie Grafu

Utworzenie grafu, w którym węzły reprezentują zadania a krawędzie ich wzajemny wpływ na siebie (zależności).



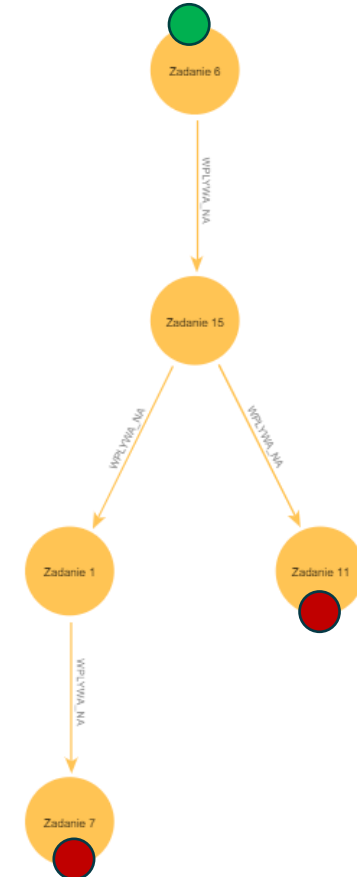
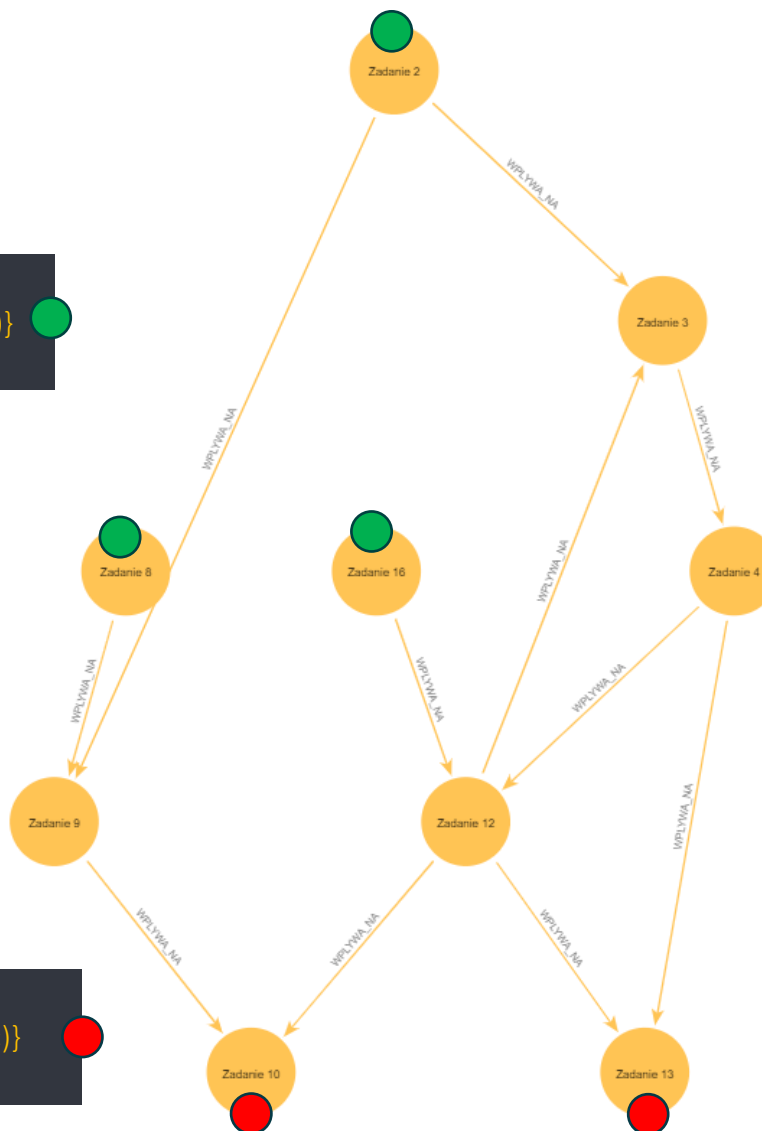
Relacyjne zapytania (Cypher)

Zadania startowe (inicjujące)
(korzenie w drzewie)

```
MATCH (korzen:Zadanie)
WHERE NOT EXISTS {MATCH (korzen)-[:WPLYWA_NA]-(rodzic)}
RETURN korzen
```

Zadania końcowe (cele)
(liście w drzewie)

```
MATCH (lisc:Zadanie)
WHERE NOT EXISTS {MATCH (dziecko)-[:WPLYWA_NA]-(lisc)}
RETURN lisc
```

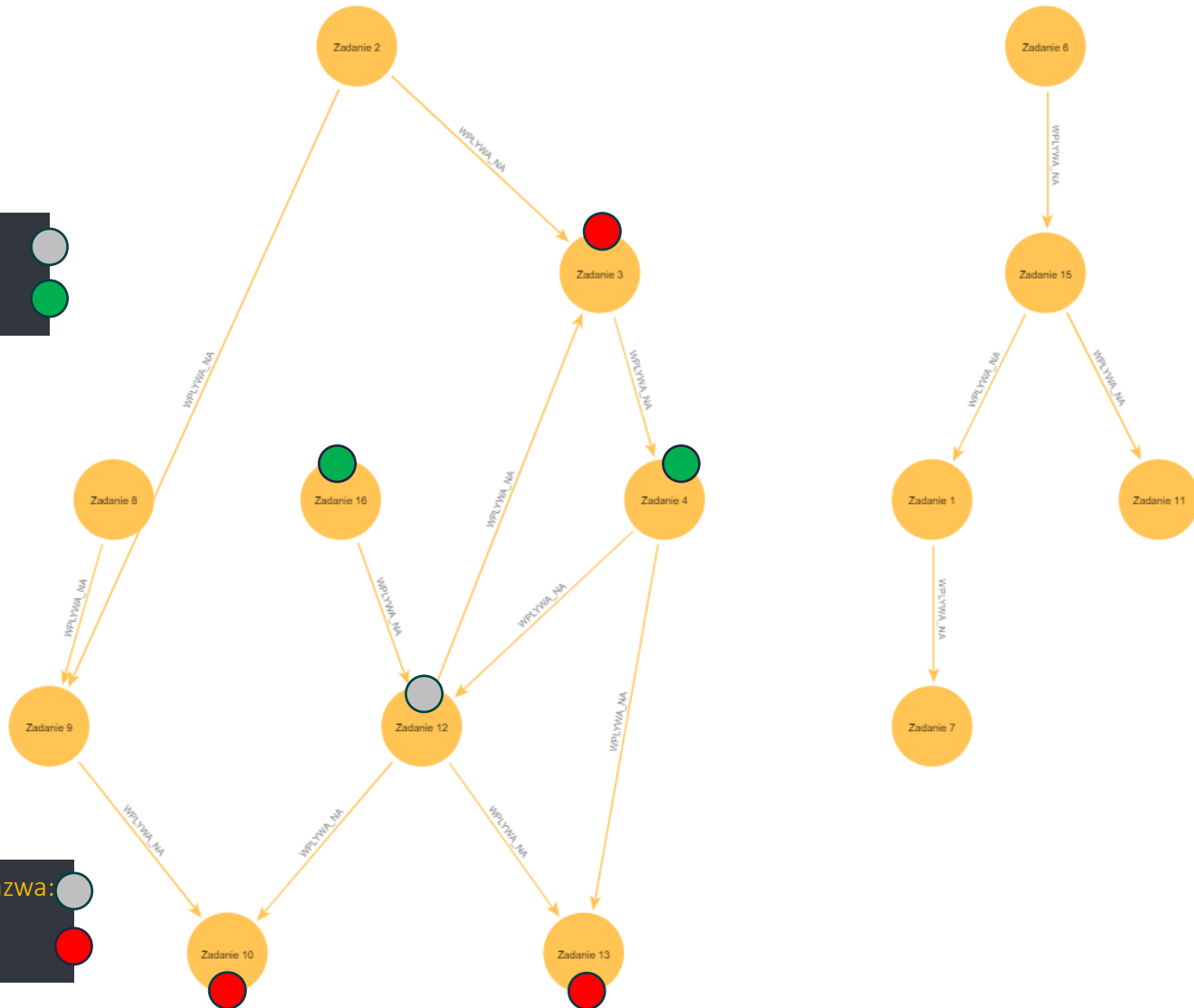


Relacyjne zapytania (Cypher)

Jakie zadania wpływają bezpośrednio na zadanie X
(wszystkie poprzedniki zadania X)

```
MATCH (x:Zadanie {nazwa: 'Zadanie 12'})<-[:WPLYWA_NA]-
(poprzednik:Zadanie)
RETURN x, poprzednik
```

Na jakie zadania wpływa bezpośrednio zadanie X
(wszystkie następniki zadania X)



Relacyjne zapytania (Cypher)

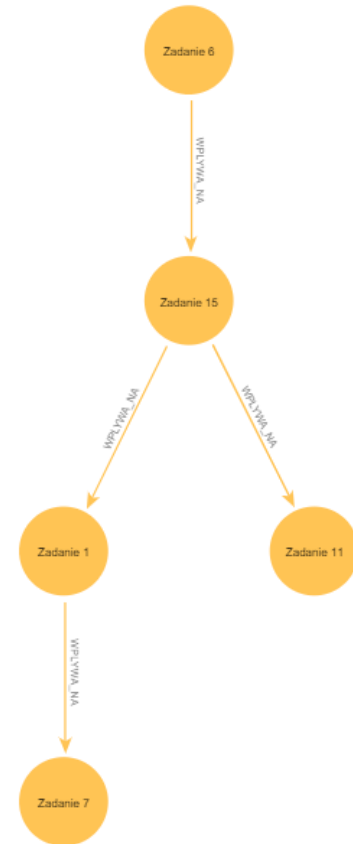
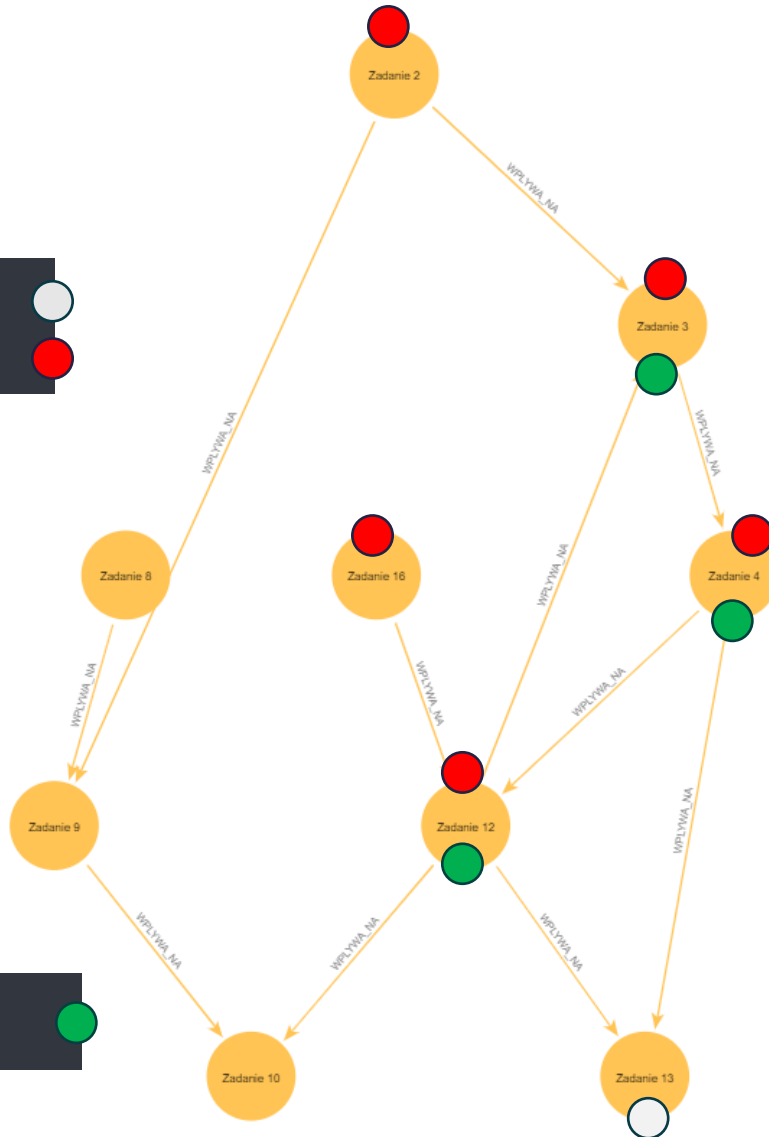
Harmonogram osiągnięcia celu X
(drogi dochodzące do zadania X)

```
MATCH (nastepnik:Zadanie)-[:WPLYWA_NA*]-(x:Zadanie  
{nazwa: 'Zadanie 13'})  
RETURN x, nastepnik
```

Cykl Sprzężenia Zwrotnego Zadań
(cykle w grafie)

Aspekt **pozytywny** i **negatywny**

```
MATCH path = (n:Zadanie)-[:WPLYWA_NA*]->(n:Zadanie)  
RETURN path
```



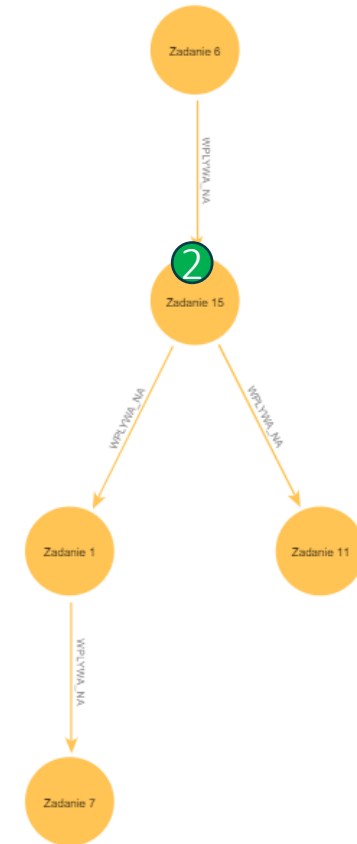
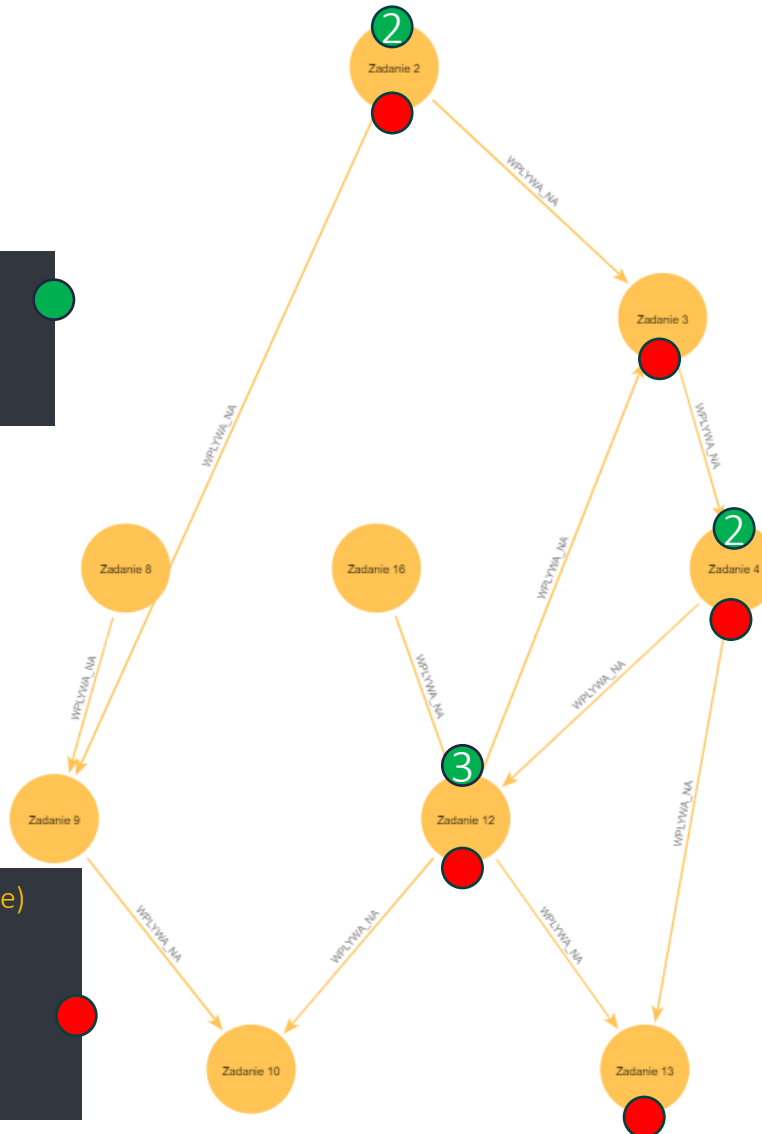
Relacyjne zapytania (Cypher)

Kluczowe zadania / krytyczne
(stopnie wychodzące węzłów)

```
MATCH ()<-[r:WPLYWA_NA]-(n:Zadanie)
WITH n, COUNT(r) as outDegree
RETURN n, outDegree
ORDER BY outDegree DESC
```

Ścieżka krytyczna
(bezpośredni wpływ na całkowity
czas trwania projektu)

```
MATCH path=(start:Zadanie)-[:WPLYWA_NA*]->(end:Zadanie)
WITH path, REDUCE(s = 0, n IN nodes(path) | s +
n.czas_trwania) AS totalDuration
RETURN path AS CriticalPath, totalDuration
ORDER BY totalDuration DESC
LIMIT 1
```



Porównanie Cypher z SQL

Powiązanie **zadań** z **procesem A** i realizującymi je **osobami** pracującymi na **stanowiskach** w **komórkach** wewnętrznych organizacji

Cypher

```
MATCH d=(n:Proces {nazwa: 'Proces A'})-[:ZAWIERA_ZADANIE]->()-[:REALIZOWANE_PRZEZ]->()-[:PRACUJE_JAKO]->()-[:JEST_W]->(k)
RETURN d
```

SQL

```
SELECT * FROM Proces
INNER JOIN Zadanie ON Proces.id = Zadanie.idProces
INNER JOIN Osoba ON Zadanie.idOsoba = Osoba.id
INNER JOIN Stanowisko ON Osoba.idStanowisko = Stanowisko.id
INNER JOIN Komorka ON Stanowisko.idKomorka = Komorka.id
WHERE Proces.nazwa = 'Proces A';
```

Warunek: związki tylko 1:n



Detekcja liderów

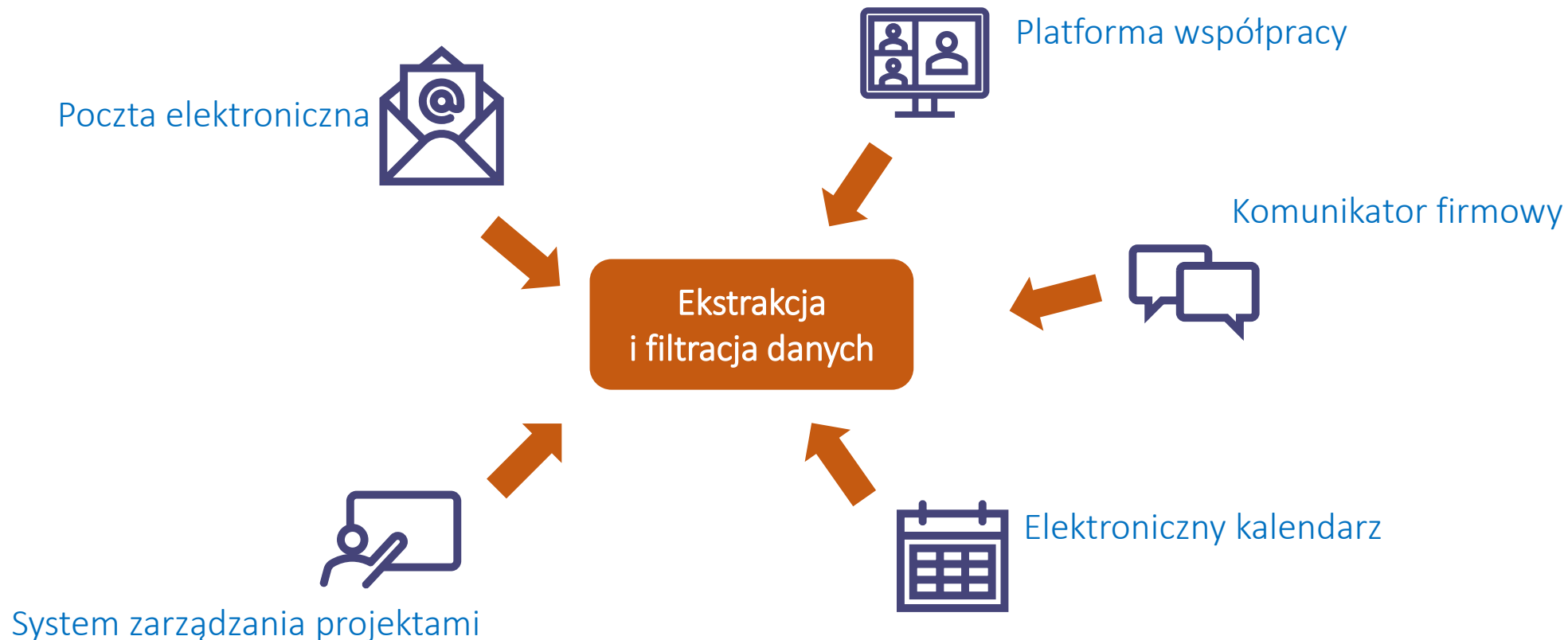
Zidentyfikować nieformalnych liderów w organizacji, którzy mogą nie zajmować oficjalnych stanowisk kierowniczych, ale mają znaczący wpływ na morale, przepływ informacji oraz produktywność.



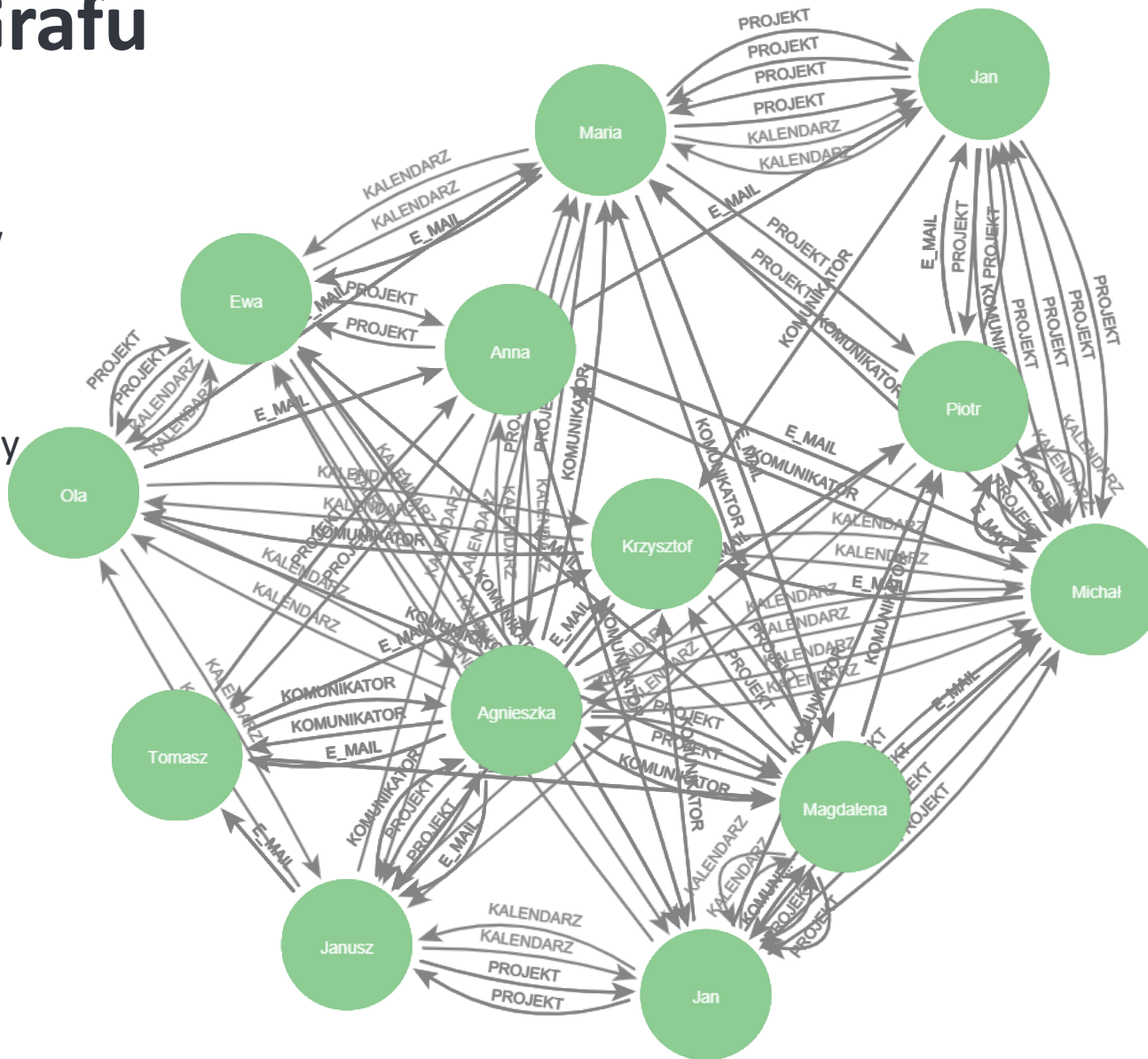
Detekcja Liderów

Eksploracja Danych (Data Mining)

Zebranie danych o interakcjach i komunikacji między pracownikami organizacji.



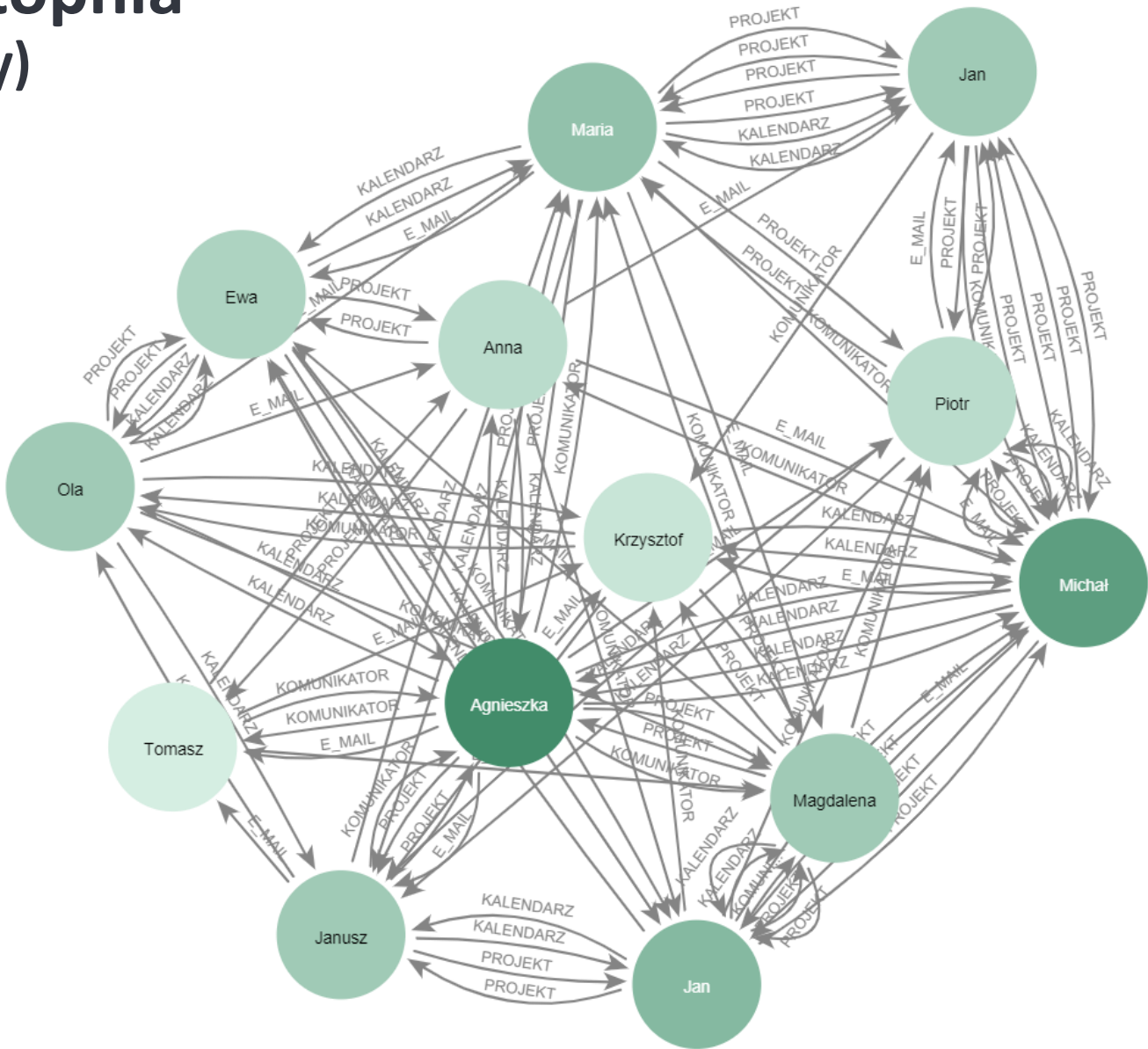
Utworzenie grafu, w którym węzły reprezentują osoby (pracowników), a krawędzie różne typy interakcji i komunikacji między pracownikami organizacji.



Centralność stopnia (Degree Centrality)

To miara, która określa, jak wiele połączeń (krawędzi) prowadzi do danego wężła (osoby).

Wskazuje na to, jak „popularna” jest dana osoba.



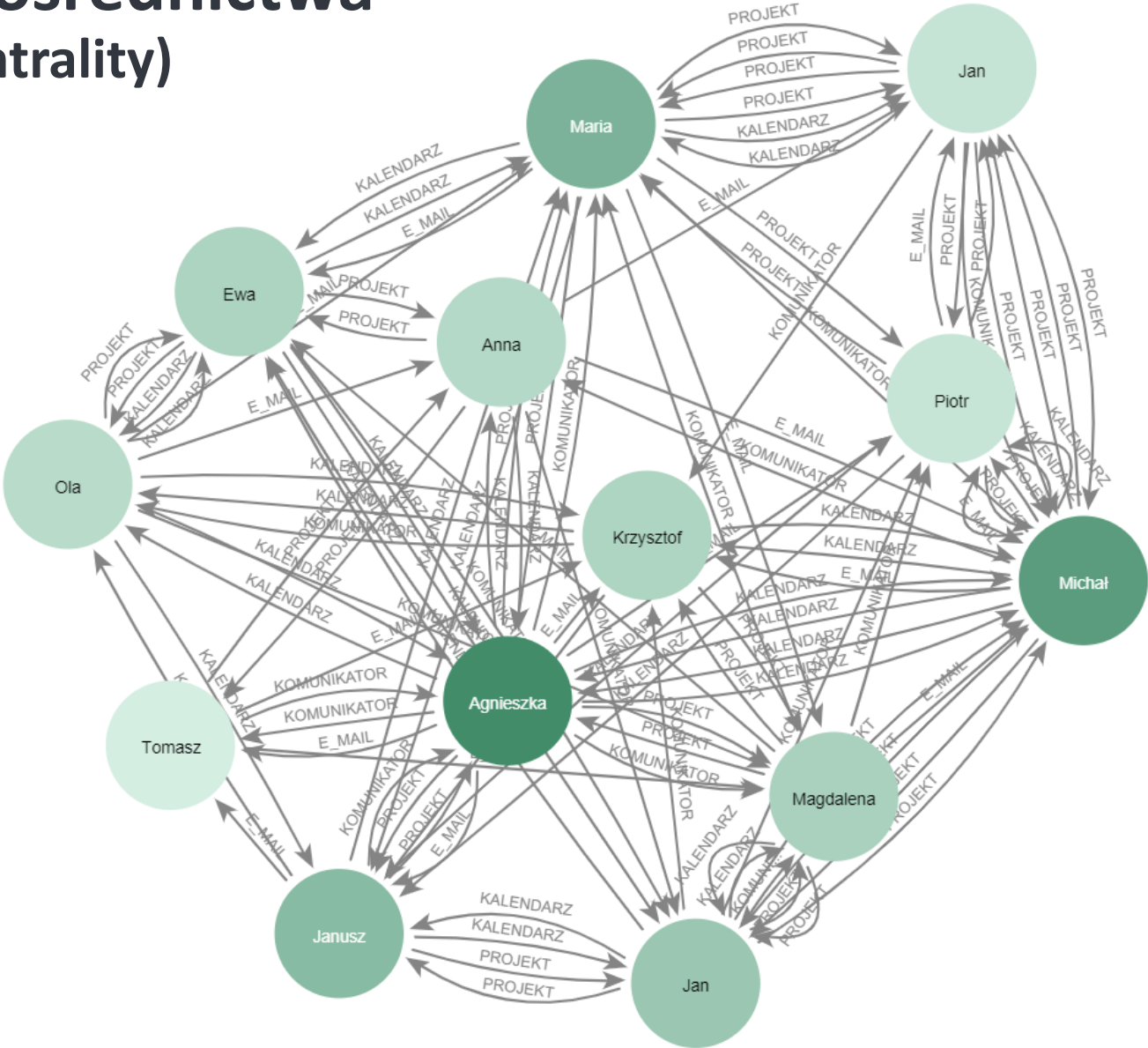
Pozycja	ID	Osoba	Wynik	
1	25	"Agnieszka"	15,00	1,25000
2	24	"Michał"	13,00	1,08333
3	41	"Jan"	10,00	0,83333
4	34	"Maria"	9,00	0,75000
5	21	"Magdalena"	8,00	0,66667
6	26	"Janusz"	8,00	0,66667
7	31	"Jan"	8,00	0,66667
8	40	"Ola"	8,00	0,66667
9	23	"Ewa"	7,00	0,58333
10	32	"Anna"	6,00	0,50000
11	33	"Piotr"	6,00	0,50000
12	22	"Krzysztof"	5,00	0,41667
13	35	"Tomasz"	4,00	0,33333

Centralność pośrednictwa

(Betweenness centrality)

To miara, która wskazuje na stopień, w jakim dany węzeł leży na ścieżkach między innymi węzłami.

Wskazuje na to jak ważna jest osoba dla ciągłości kanałów przepływu informacji.



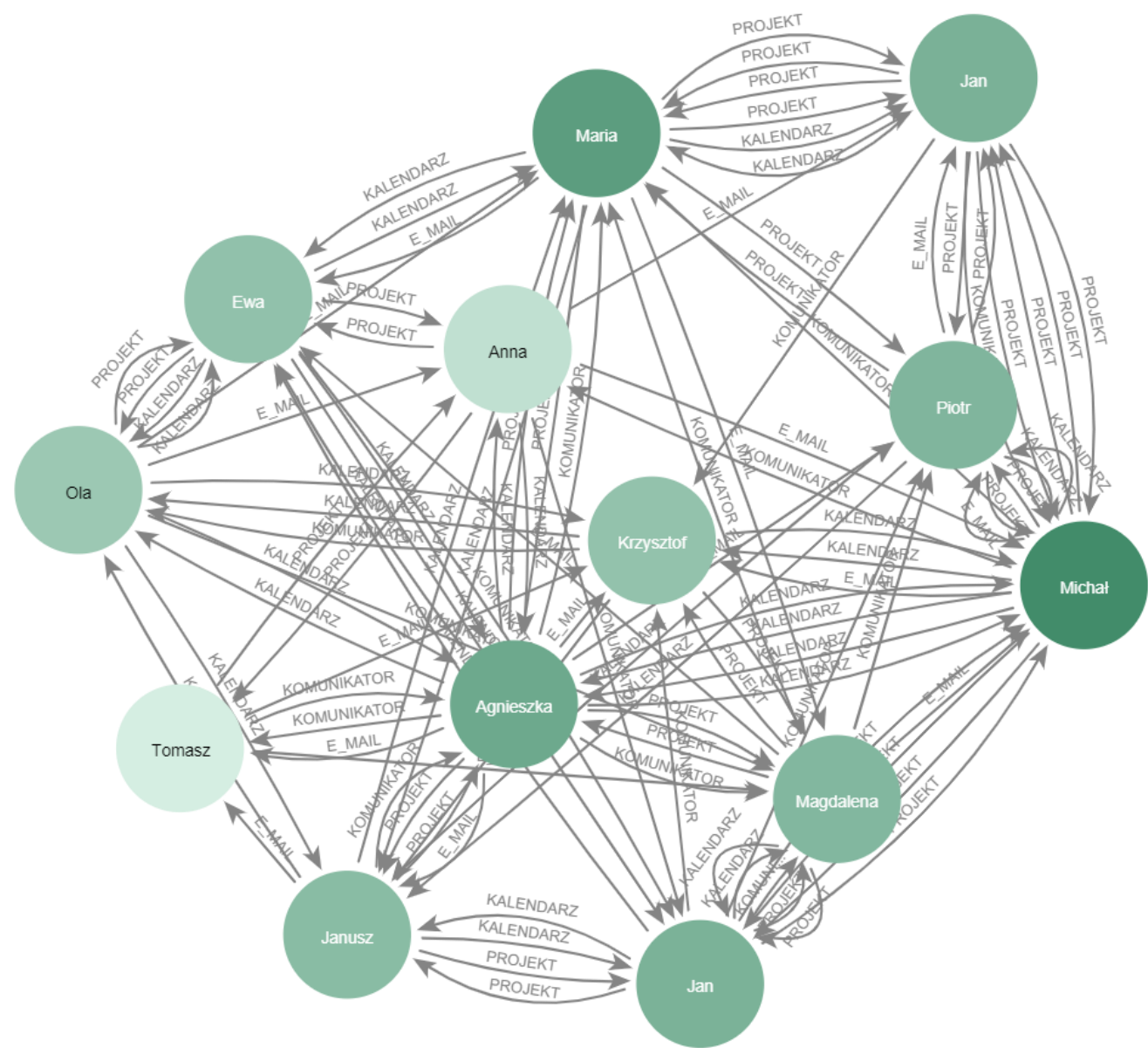
Pozycja	ID	Osoba	Wynik
1	25	"Agnieszka"	15,40300
2	24	"Michał"	12,95922
3	34	"Maria"	9,73657
4	26	"Janusz"	8,45818
5	41	"Jan"	6,71694
6	21	"Magdalena"	5,15317
7	23	"Ewa"	4,81508
8	22	"Krzysztof"	4,69628
9	32	"Anna"	4,00580
10	40	"Ola"	3,70001
11	31	"Jan"	2,33810
12	33	"Piotr"	2,19072
13	35	"Tomasz"	0,82692

PageRank

(PageRank)

To miara wpływu węzła w sieci ze względu na jakość jego połączeń.

Pokazuje jak ważna jest dana osoba ze względu na wagę połączeń z innymi ważnymi osobami.



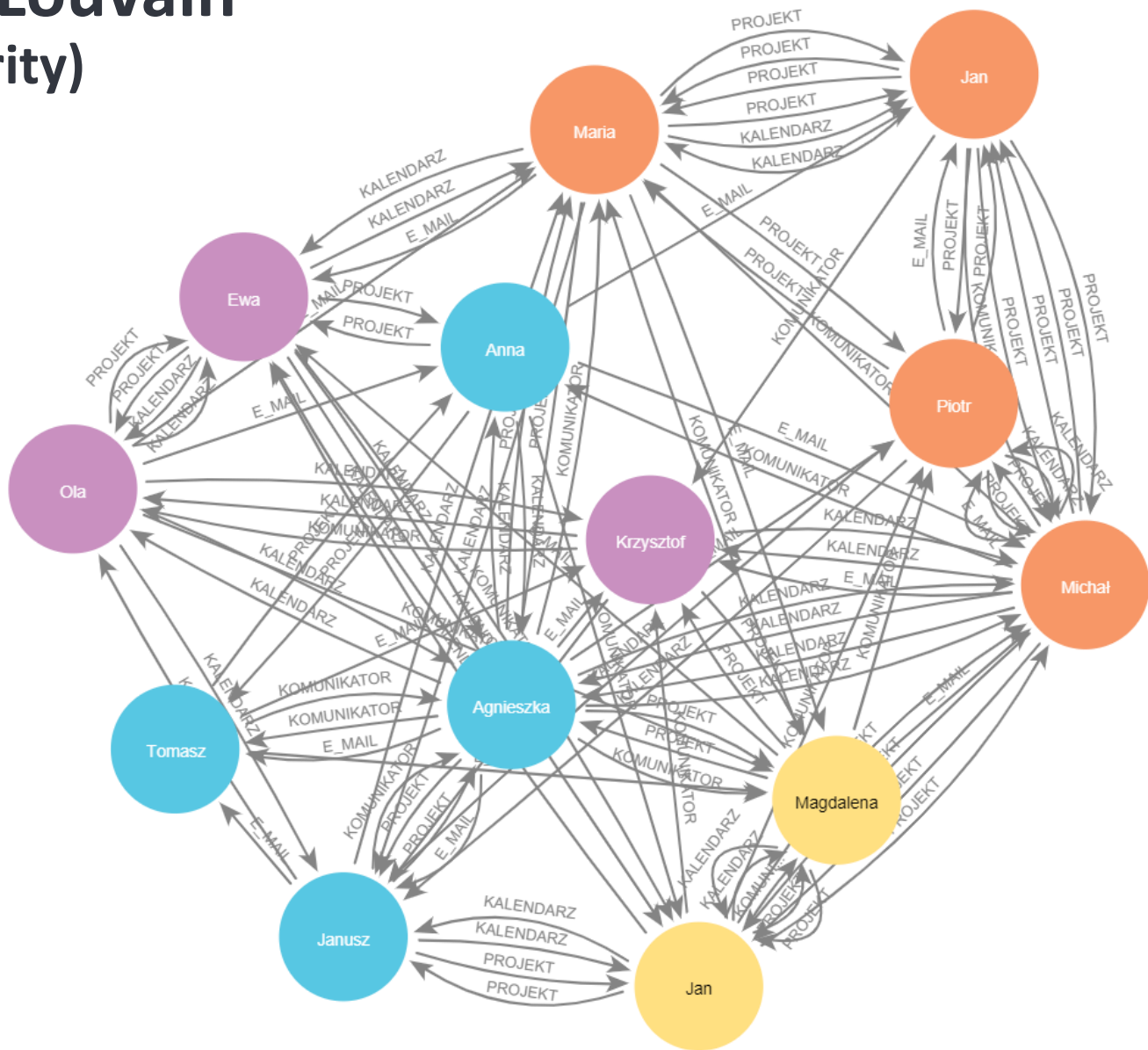
Pozycja	ID	Osoba	Wynik
1	24	"Michał"	1,40721
2	34	"Maria"	1,24306
3	25	"Agnieszka"	1,13165
4	31	"Jan"	1,04978
5	41	"Jan"	1,04315
6	33	"Piotr"	1,00760
7	21	"Magdalena"	0,99239
8	26	"Janusz"	0,94848
9	23	"Ewa"	0,89871
10	22	"Krzysztof"	0,88480
11	40	"Ola"	0,82921
12	32	"Anna"	0,59842
13	35	"Tomasz"	0,46166

Modułowość Louvain

(Louvain Modularity)

Jest stosowany do wykrywania społeczności w dużych sieciach.

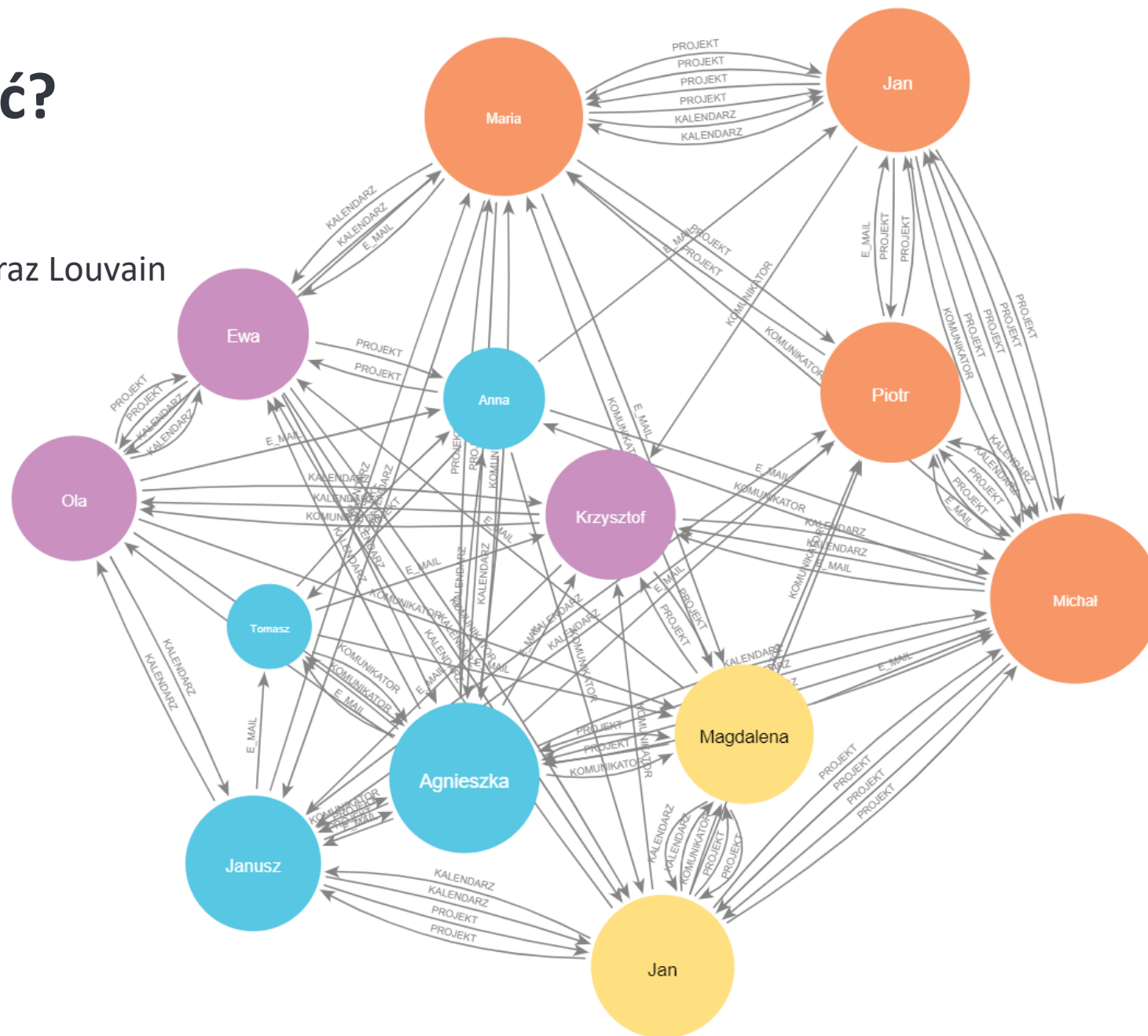
Identyfikuje grupy o podobnych zainteresowaniach lub wzorcach interakcji.



ID	Osoba	Grupa
22	"Krzysztof"	11
23	"Ewa"	11
40	"Ola"	11
25	"Agnieszka"	10
26	"Janusz"	10
32	"Anna"	10
35	"Tomasz"	10
24	"Michał"	9
31	"Jan"	9
33	"Piotr"	9
34	"Maria"	9
21	"Magdalena"	0
41	"Jan"	0

Czy i kogo należy się bać?

Połączenie wyników:
algorytmy PageRanking oraz Louvain

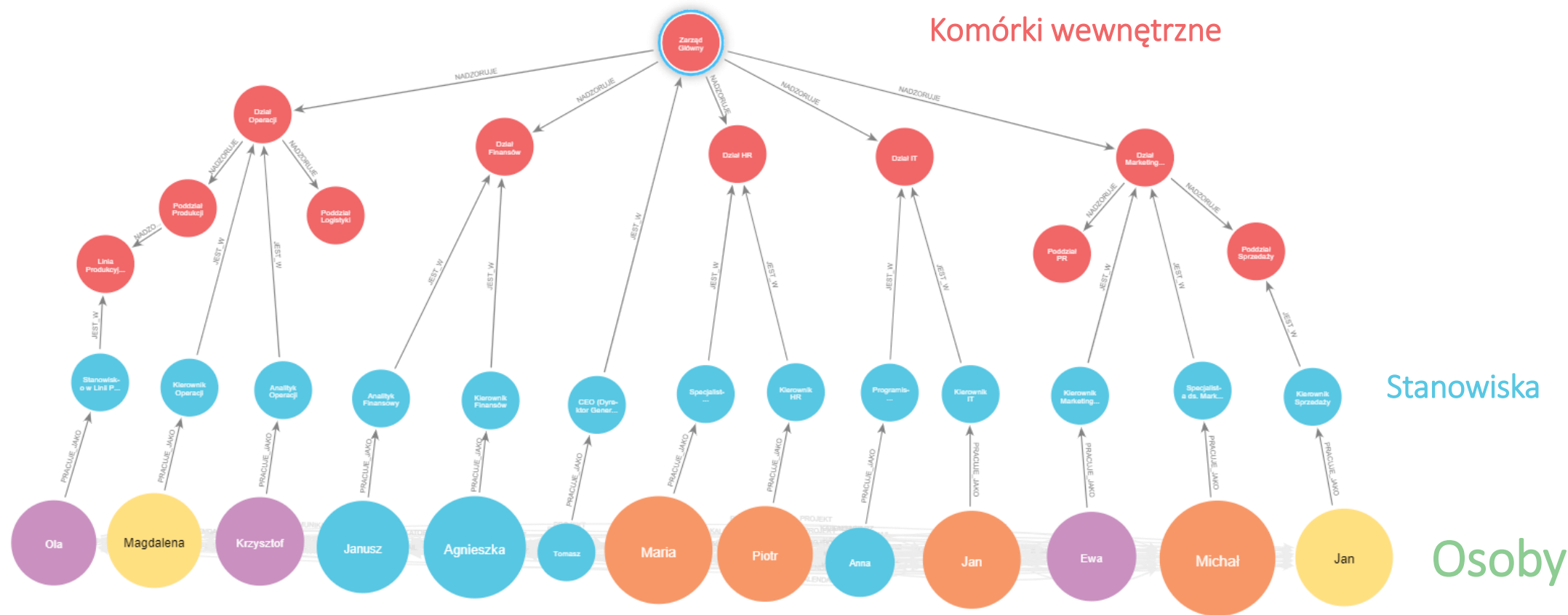


Czy i kogo należy się bać?

Ranking osób
pod względem
rozpatrywanych miar

Louvain	Pozycja			Średnia
	Degree	Betweenness	PageRank	
"Agnieszka"	1	1	3	1,67
"Michał"	2	2	1	1,67
"Maria"	4	3	2	3,00
"Jan"	3	5	5	4,33
"Magdalena"	5	6	7	6,00
"Janusz"	6	4	8	6,00
"Jan"	7	11	4	7,33
"Ewa"	9	7	9	8,33
"Ola"	8	10	11	9,67
"Piotr"	11	12	6	9,67
"Krzysztof"	12	8	10	10,00
"Anna"	10	9	12	10,33
"Tomasz"	13	13	13	13,00

Graf liderów organizacji





Algorytmy grafowe

ANALIZA DANYCH
Z WYKORZYSTANIEM BAZY
GRAFOWEJ Neo4J
I ALGORYTMÓW GRAFOWYCH

Algorytmy grafowe



Algorytmy wyszukiwania ścieżek

Wyszukują optymalne ścieżki pomiędzy węzłami dla określonych kryteriów



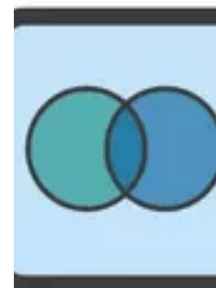
Algorytmy centralności

Pozwalają ocenić, jakie węzły w sieci są najważniejsze, czyli mogą mieć największy wpływ



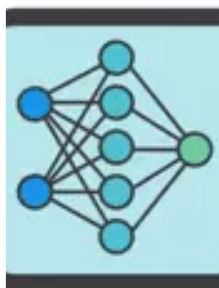
Algorytmy wspólnoty

Identyfikują skupiska węzłów, które są gęściej połączone między sobą niż z resztą sieci



Algorytmy podobieństwa

Obliczają stopień podobieństwa między elementami



Algorytmy detekcji anomalii

Wykorzystywane są do identyfikowania nieoczekiwanych wzorców lub obserwacji w danych



Algorytmy predykcji relacji

Prognozują czy dwa węzły w grafie mogą w przyszłości stworzyć połączenie

Algorytmy wyszukiwania ścieżek

Algorytmy wyszukiwania ścieżek

Algorytmy wyszukiwania ścieżek znajdują ścieżkę między dwoma lub więcej węzłami lub oceniają dostępność i jakość ścieżek.

- Znalezienie ścieżki: Lokalizowanie jednej lub kilku ścieżek spełniających określone kryteria, takie jak minimalny koszt, najkrótszy czas przejścia lub najmniejsza liczba przejść między wierzchołkami.
- Ocena ścieżki: Ocenianie dostępności i jakości ścieżek na podstawie różnych parametrów, jak dystans, przepustowość czy koszt.



- Nawigacja do wyznaczania najkrótszej trasy między punktami.
- Planowanie tras w logistyce do optymalizacji kosztów dostaw.

Algorytmy wyszukiwania ścieżek

<https://neo4j.com/docs/graph-data-science/current/algorithms/pathfinding/>

Algorytm	Charakterystyka	Zastosowanie
Delta-Stepping Single-Source Shortest Path	Równoległy algorytm znajdujący najkrótsze ścieżki z pojedynczego źródła.	W dużych grafach wymagających równoległego przetwarzania.
Dijkstra Source-Target Shortest Path	Znajduje najkrótszą ścieżkę między parą wierzchołków w grafie ważonym z nieujemnymi wagami.	W systemach nawigacji i mapowania tras między dwoma punktami.
Dijkstra Single-Source Shortest Path	Oblicza najkrótsze ścieżki od jednego wierzchołka do wszystkich innych w grafie.	Gdy potrzebna jest tabela tras z jednego punktu do wielu celów.
A* Shortest Path	<u>Heurystyczny</u> algorytm używający funkcji kosztów do szacowania najbardziej obiecujących ścieżek.	W grach lub robotyce do planowania ruchu z uwzględnieniem kosztów przemieszczenia.
Yen's k-Shortest Path Algorithm	Wariant Dijkstry znajdujący do K różnych najkrótszych ścieżek między parą wierzchołków.	Wyznaczanie różnych alternatywnych tras, np. w planowaniu redundancji sieci.
Breadth First Search (BFS)	Przeszukuje wszerek, używany do znajdowania najkrótszej ścieżki w grafach nieważonych.	Wykrywanie wszystkich połączeń na określonym poziomie odległości od punktu początkowego.
Depth First Search (DFS)	Przeszukuje w głąb, używany do eksploracji wszystkich ścieżek aż do osiągnięcia końca ścieżki.	Sprawdzanie spójności grafu, znajdowanie składowych spójnych, analiza problemów wymagających kompleksowej eksploracji.
Random Walk	Generuje losowe ścieżki w grafie na zasadzie losowości.	Symulacja rozprzestrzeniania informacji w sieciach społecznościowych lub algorytmach rekomendacji.
Bellman-Ford Single-Source Shortest Path	Radzi sobie z negatywnymi wagami krawędzi, znajdując najkrótszą ścieżkę.	Używany w grafach, które mogą zawierać negatywne wagi, np. do uwzględniania różnego rodzaju kosztów.
Minimum Weight Spanning Tree (MST)	Tworzy minimalne drzewo rozpinające dla grafu ważonego, minimalizując sumaryczny koszt.	W problemach optymalizacyjnych, takich jak projektowanie sieci komunikacyjnych, elektrycznych czy hydraulicznych.

Algorytmy wyszukiwania ścieżek

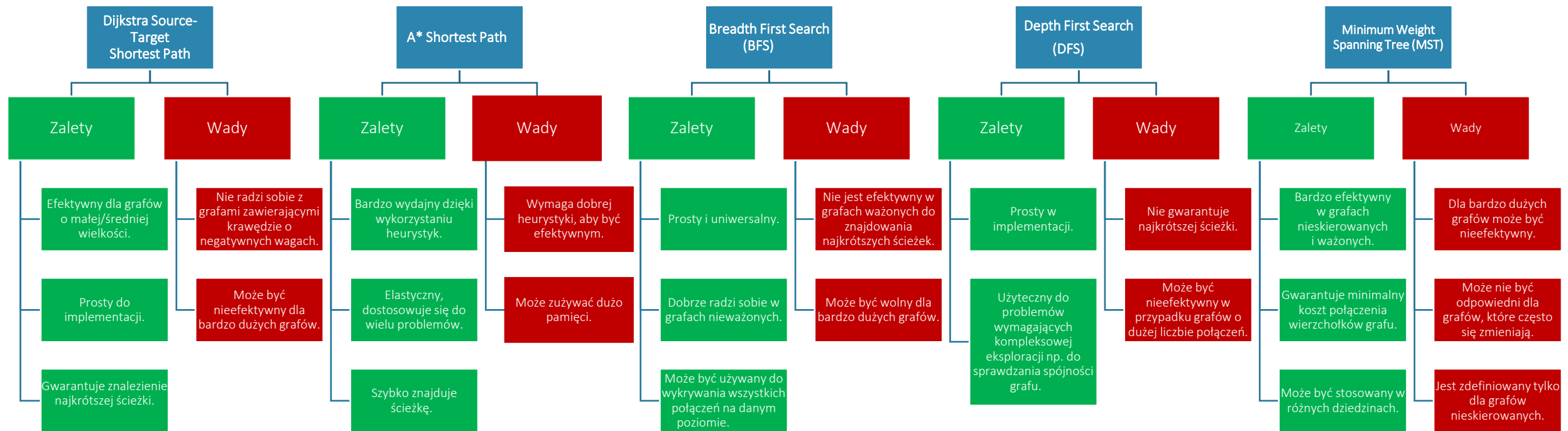
Heurystyka w kontekście algorytmów, takich jak A*,

to metoda umożliwiająca ocenę najbardziej obiecujących kierunków przeszukiwania przestrzeni stanów, aby znaleźć rozwiązanie. Jest to rodzaj "intuicji" lub "przewidywania", które pomagają algorytmowi określić, które węzły w grafie powinny być eksplorowane jako pierwsze, aby zwiększyć szansę na szybkie dotarcie do celu.

Heurystyka w algorytmie A* często przyjmuje formę funkcji kosztu, która ocenia odległość (niekoniecznie geograficzną) od danego punktu w grafie do punktu docelowego. Jeśli jest dobrze zaprojektowana, pozwala znacząco przyspieszyć proces wyszukiwania w porównaniu do algorytmów nieheurystycznych, ponieważ zmniejsza liczbę węzłów, które algorytm musi rozważyć.

Przykładem funkcji heurystycznej może być prosty dystans euklidesowy w problemach nawigacyjnych, gdzie oceniamy bezpośrednią odległość między dwoma punktami bez uwzględniania rzeczywistych przeszkód. Heurystyka powinna być możliwie jak najdokładniejszym oszacowaniem rzeczywistego kosztu dojścia do celu, nie przeszacowując go.

Algorytmy wyszukiwania ścieżek



Algorytmy wyszukiwania ścieżek

Ćwiczenie: Optymalizacja trasy dostaw w firmie kurierskiej

Założmy, że pracujesz w firmie kurierskiej jako analityk systemów i Twoim zadaniem jest zminimalizowanie czasu i kosztów dostaw. Twoja firma dysponuje rozbudowaną siecią połączeń między hubami logistycznymi w różnych miastach. Każde połączenie między hubami ma określony koszt i czas przejazdu, które mogą się zmieniać w zależności od warunków drogowych, pogodowych i natężenia ruchu. Twoim zadaniem jest stworzenie systemu, który będzie w stanie w czasie rzeczywistym wyznaczać najbardziej optymalne trasy dla kurierów.

Zadanie

1. Przeanalizuj dostępne algorytmy wyszukiwania ścieżek i wybierz najbardziej odpowiedni do zastosowania w opisanej sytuacji.
2. Uzasadnij wybór algorytmu, biorąc pod uwagę specyfikę problemu, taką jak dynamika zmian i rozmiar sieci połączeń.

Pamiętaj, że powinieneś wziąć pod uwagę nie tylko efektywność algorytmu w znajdowaniu najkrótszej ścieżki, ale także jego zdolność do radzenia sobie z dużą liczbą wierzchołków i dynamicznie zmieniającymi się wagami krawędzi.

Algorytmy wyszukiwania ścieżek

Propozycja odpowiedzi

Dla zadania optymalizacji tras dostaw, gdzie czynniki takie jak czas i koszty mogą się dynamicznie zmieniać, algorytm A* wydaje się być najlepszym wyborem. A* oferuje połączenie wydajności algorytmu Dijkstry z elastycznością dodatkowych heurystyk, które pomagają przewidywać koszt dojścia do celu, znacząco redukując obszar przeszukiwania grafu.

Dzięki możliwości wykorzystania funkcji heurystycznej algorytm A* jest szczególnie odpowiedni do dynamicznych środowisk, takich jak logistyka, gdzie czas i koszty dostaw mogą się zmieniać. Algorytm ten pozwala na szybkie reagowanie na takie zmiany i przeszukiwanie dużych grafów w poszukiwaniu optymalnej trasy dostawy.

Zalety:

- Szybkość działania dzięki zastosowaniu heurystyki.
- Możliwość dostosowania do specyficznych potrzeb dzięki wyborowi odpowiedniej funkcji heurystycznej.
- Skuteczność w znajdowaniu najkrótszej ścieżki, szczególnie w skomplikowanych sieciach drogowych.

Wady:

- Konieczność posiadania odpowiedniej funkcji heurystycznej dostosowanej do danego zastosowania.
- Potencjalnie duże zużycie pamięci, szczególnie w bardzo dużych grafach.

Algorytmy wspólnoty

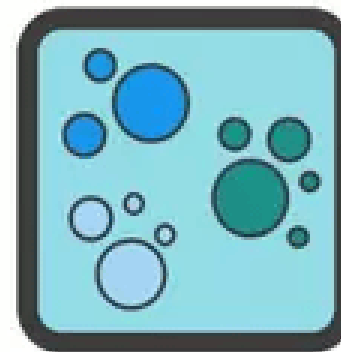
Algorytmy wspólnoty

Algorytmy wykrywania społeczności są wykorzystywane do oceny, w jaki sposób grupy węzłów są grupowane lub dzielone, a także ich tendencji do wzmacniania się lub rozpadania.

Te grupy, często nazywane wspólnotami, charakteryzują się względnie gęstszym połączeniem między węzłami wewnątrz grupy niż z węzłami zewnętrznymi.

Proces wykrywania społeczności pozwala zrozumieć nie tylko strukturę i organizację sieci, ale także może dostarczyć wglądu w sposób, w jaki informacje lub wpływy rozprzestrzeniają się w sieci.

- Analiza sieci społecznych do wykrywania grup zainteresowań lub wspólnot.
- Biologia obliczeniowa do identyfikowania funkcjonalnych klastrów w sieciach interakcji białkowych.



Algorytmy wspólnoty

<https://neo4j.com/docs/graph-data-science/current/algorithms/community/>

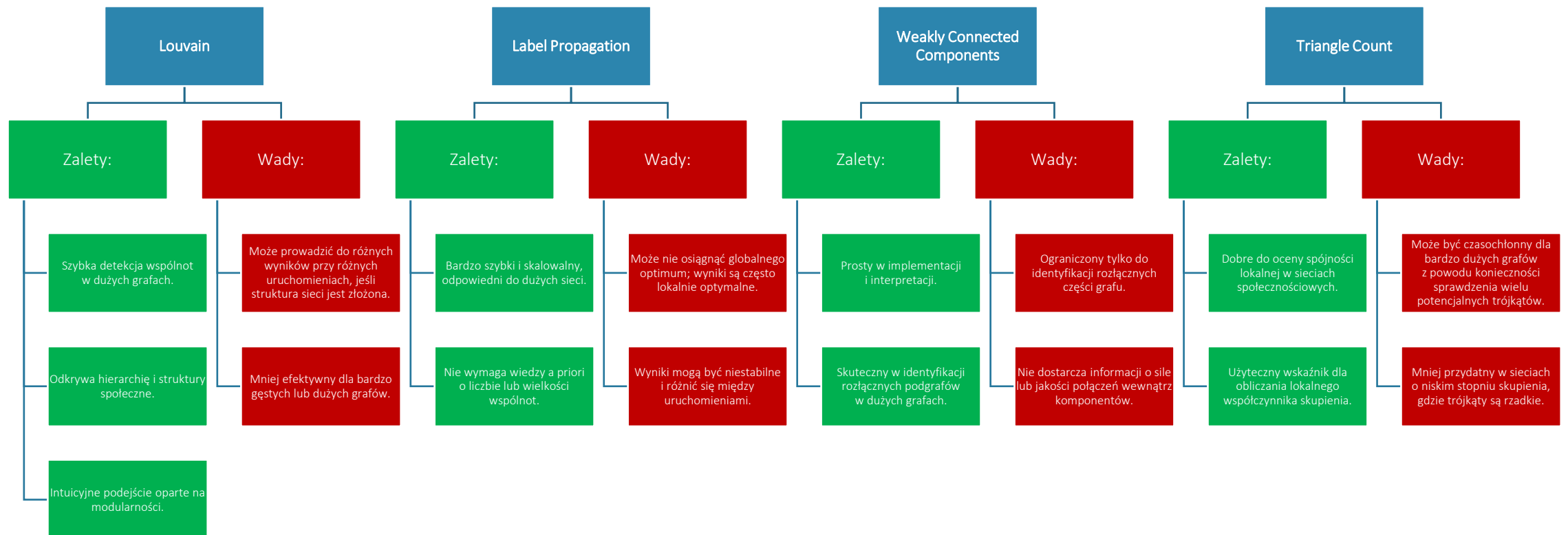
Algorytm	Charakterystyka	Zastosowanie
Conductance metric	Mierzy przepływność podziału na wspólnoty. Niskie wartości oznaczają mocne podziały.	Ocena jakości podziału grafu na wspólnoty.
K-Core Decomposition	Identyfikuje gęste podgrafy i hierarchie w sieci.	Identyfikacja silnych grup w sieciach społecznościowych lub wykrywanie hierarchii i rdzenia sieci.
K-1 Coloring	Przypisuje kolory węzłom grafu tak, aby sąsiednie węzły nie miały tego samego koloru.	Rozwiązywanie problemów alokacji zasobów, jak przydział pasma radiowego lub kolorowanie map.
K-Means Clustering	Dzieli zbiór obiektów na k klastrów na podstawie ich atrybutów.	Grupowanie węzłów w sieciach społecznościowych na podstawie podobieństwa atrybutów.
Label Propagation	Pozwala węzłom przyjmować etykietę dominującą wśród ich sąsiadów.	Analiza wspólnot w sieciach społecznościowych, gdzie etykiety reprezentują przynależność do grup.
Leiden	Udoskonalony algorytm Louvain, zapewniający dokładniejszy podział na wspólnoty.	Wykrywanie wspólnot w dużych sieciach, często skuteczniej niż tradycyjny algorytm Louvain.
Local Clustering Coefficient	Mierzy, jak bardzo sąsiedzi węzła są połączeni, wskazując na lokalną gęstość grupowania.	Analiza spójności społeczności, wykrywanie struktur społecznych o wysokiej gęstości w większych sieciach.

Algorytmy wspólnoty

<https://neo4j.com/docs/graph-data-science/current/algorithms/community/>

Algorytm	Charakterystyka	Zastosowanie
Louvain Modularity	Heurystyczna metoda oparta na optymalizacji modularności, identyfikująca wspólnoty w dużych sieciach.	Identyfikacja wspólnot w sieciach społecznościowych, biologicznych i sieciach współpracy.
Modularity metric	Mierzy siłę struktury podziału sieci na wspólnoty. Wysoka wartość wskazuje na dobrze zdefiniowane grupy.	Ocena jakości podziału sieci na wspólnoty i porównywanie różnych podziałów.
Modularity Optimization	Dostosowuje strukturę sieci w celu zwiększenia jej modularności.	Optymalizacja istniejącego podziału na wspólnoty, często w połączeniu z algorytmem Louvain lub Leiden.
Strongly Connected Components	Maksymalne podgrafy w grafie skierowanym, gdzie każda para węzłów jest osiągalna z każdego innego.	Analiza strukturalna sieci skierowanych, takich jak sieci stron internetowych lub transakcji finansowych.
Triangle Count	Oblicza liczbę trójkątów w sieci (zestawy 3 połączonych węzłów).	Pomiar tendencji do grupowania się w sieciach społecznościowych, gdzie trójkąty wskazują na silniejsze relacje.
Weakly Connected Components	Podgrafy w grafie nieskierowanym, w których istnieje ścieżka między każdą parą węzłów.	Identyfikacja izolowanych grup w sieciach, takich jak społeczności internetowe lub sieci sensorów.

Algorytmy wspólnoty



Algorytmy wspólnoty

Ćwiczenie: Analiza Struktury Społeczności w Forum Korporacyjnym

Założmy, że jesteś analitykiem danych w dużej korporacji i masz za zadanie zrozumieć, jak pracownicy wchodzą w interakcje na firmowym forum dyskusyjnym. Chcesz odkryć istniejące wspólnoty, zrozumieć, jak są one ze sobą powiązane oraz zidentyfikować liderów myśli i najbardziej wpływowe osoby wewnątrz tych społeczności.

Zadanie

1. Przeanalizuj dostępne algorytmy wspólnoty i wybierz najbardziej odpowiedni do zastosowania w opisanej sytuacji.
2. Uzasadnij wybór algorytmu, biorąc pod uwagę specyfikę problemu, taką jak dynamika zmian i rozmiar sieci połączeń.

Algorytmy wspólnoty

Propozycja odpowiedzi

Algorytm Louvain jest dobrze dostosowany do dużych sieci społecznościowych dzięki swojej skuteczności w identyfikacji hierarchicznej struktury wspólnot i zdolności do odkrywania wzajemnych powiązań między nimi. Może pomóc w odkrywaniu zarówno małych, blisko powiązanych grup dyskusyjnych, jak i szerszych trendów w komunikacji korporacyjnej.

Zalety:

- Zdolność do pracy z dużymi zestawami danych.
- Efektywne wykrywanie społeczności o różnych rozmiarach.
- Możliwość identyfikacji wpływowych użytkowników wewnątrz i między grupami.

Wady:

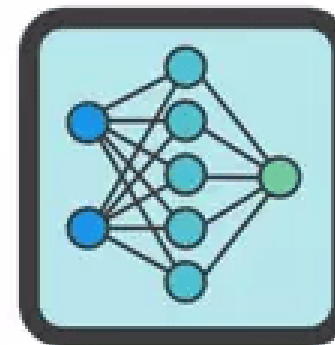
- Wyniki mogą się różnić w zależności od wykonania, co wymaga wielokrotnych uruchomień do weryfikacji wyników.
- Algorytm może nie radzić sobie dobrze z bardzo dynamicznie zmieniającymi się strukturami sieci.

Algorytmy detekcji anomalii

Algorytmy detekcji anomalii

są narzędziami służącymi do identyfikacji nieoczekiwanych wzorców lub obserwacji w danych, które mogą wskazywać na błędy, oszustwa, awarie systemowe lub inne ważne zdarzenia.

Takie anomalie mogą wskazywać na różnorodne kwestie, od błędów po podejrzaną celową aktywność.



- Wykrywanie oszustw finansowych poprzez identyfikację nietypowych wzorców transakcji.
- Monitoring sieci komputerowych w celu wychwycenia nietypowego ruchu wskazującego na potencjalne ataki cybernetyczne.

Algorytmy detekcji anomalii

Nie mają bezpośredniej implementacji w Neo4j

Korzystanie z wtyczek:

- **Anomaly Detection Library:** Ta wtyczka implementuje kilka algorytmów detekcji anomalii, takich jak Local Outlier Detection (LoF) i Isolation Forest.
<https://github.com/neo4j>
- **Neo4j Anomaly Detection:** Ta wtyczka implementuje algorytm oparty na podejściu opartym na grafie do wykrywania anomalii.
<https://github.com/topics/graph-anomaly-detection>

Korzystanie z zapytań Cypher:

- Można użyć zapytań Cypher do napisania własnych algorytmów detekcji anomalii.
Na przykład można użyć zapytania Cypher, aby znaleźć węzły lub relacje, które mają nietypowe właściwości lub schematy połączeń.

Algorytmy detekcji anomalii

Algorytm	Opis	Zastosowanie w wykrywaniu oszustw
Lokalizacja odstających węzłów (LoF)	Identyfikuje węzły o nietypowej liczbie połączeń lub innych atrybutach.	Klienci, konta lub sprzedawcy z nietypową aktywnością transakcyjną.
Las izolacji	Buduje losowe drzewa decyzyjne, aby identyfikować anomalie.	Klienci lub transakcje, które nie pasują do typowych wzorców.
Wykrywanie społeczności	Identyfikuje grupy powiązanych węzłów.	Siatki oszustów lub inne grupy osób zaangażowane w podejrzone działania.
Wykrywanie ścieżek	Śledzi przepływ wartości lub informacji w grafie.	Nietypowe wzorce przepływu pieniędzy np. przelewy pomiędzy klientami /kontami poprzez „pośredników”.
Centralność węzła	Mierzy względną ważność węzłów w grafie.	Klienci, konta lub sprzedawcy odgrywający kluczową rolę w sieciach oszustów.
Odchylenie standardowe	Wykorzystuje odchylenie standardowe do identyfikacji wartości odstających w normalnie rozłożonych danych.	Nietypowe wydatki (transakcje o dużej kwocie), nietypowej aktywności (nagłe zakupy, częste wypłaty) i monitorowania trendów w danych transakcyjnych.
Klastrowanie DBSCAN	Klastruje dane i identyfikuje anomalie jako punkty danych nie należące do żadnego klastra.	Identyfikacja grup użytkowników, którzy wykazują nietypową aktywność transakcyjną, taką jak nagłe i duże wydatki.
Błąd rekonstrukcji w autoenkoderach	Porównuje oryginalne dane z ich zrekonstruowanymi wersjami przez autoenkoder.	Wykrywanie nietypowych wzorców w danych transakcyjnych, takich jak fałszywe transakcje lub nadużycia kart.
Jednoklasowy SVM (One-Class SVM)	Trenuje model SVM na danych normalnych i identyfikuje anomalie jako punkty odbiegające od modelu.	Identyfikacja użytkowników, którzy wykazują nietypową aktywność transakcyjną.

Algorytmy detekcji anomalii

Ćwiczenie: Analiza Anomalii w Transakcjach Finansowych z Użyciem Odchylenia Standardowego

Założmy, że mamy zbiór transakcji finansowych i chcemy wykryć anomalie, korzystając z odchylenia standardowego.

Podejście mogłoby wyglądać następująco:

- 1. Zbieranie danych:** Przygotowanie zestawu danych (grafu) zawierających kwoty transakcji dla każdego klienta.
- 2. Obliczenia statystyczne:** Dla każdego klienta oszacowanie średniej wartości transakcji oraz odchylenia standardowego. W Neo4j możesz to zrobić za pomocą Cypher, agregując transakcje po klientach.
- 3. Identyfikacja anomalii:** Określenie anomalii jako transakcje, które mają wartość przekraczającą określoną liczbę odchyień standardowych od średniej, na przykład:

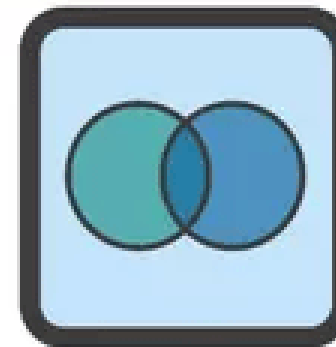
```
MATCH (c:Client)-[t:TRANSACTION]->()
WITH c, AVG(t.amount) AS avgAmount, STDEV(t.amount) AS stdevAmount
MATCH (c)-[t:TRANSACTION]->()
WHERE ABS(t.amount - avgAmount) > (2 * stdevAmount)
RETURN c, t
```

Algorytmy podobieństwa

Algorytmy podobieństwa

służą do oceny, jak bardzo podobne są do siebie węzły na podstawie określonych cech lub wzorców połączeń.

Te algorytmy wykorzystują różne metryki, takie jak współczynniki korelacji, odległość euklidesowa, czy wspólne sąsiedztwo, aby kwantyfikować podobieństwo.



- Systemy rekomendacyjne, które sugerują produkty podobne do tych, które klient już przeglądał.
- Analiza dokumentów prawnych lub naukowych w celu znalezienia podobnych przypadków lub publikacji.

Algorytmy podobieństwa

<https://neo4j.com/docs/graph-data-science/current/algorithms/similarity/>

Algorytm	Charakterystyka	Zastosowanie
Node Similarity	Algorytm obliczający podobieństwo między parami węzłów na podstawie wspólnych sąsiedztw lub cech.	Używany do rekomendacji społecznościowych, identyfikowania podobnych klientów w usługach lub produktach.
Filtered Node Similarity	Wariant algorytmu podobieństwa węzłów, który filtruje pary na podstawie określonych kryteriów.	Stosowany do wyrafinowanych systemów rekomendacyjnych, gdzie brane są pod uwagę tylko niektóre typy połączeń.
K-Nearest Neighbors	Znajduje K najbliższych sąsiadów dla danego węzła, opierając się na metrykach takich jak <u>odległość euklidesowa</u> .	Wykorzystywany w klasteryzacji, klasyfikacji danych i w systemach rekomendacji opartych na podobieństwie.
Filtered K-Nearest Neighbors	Wersja algorytmu K-Nearest Neighbors, która pozwala na zastosowanie filtrów do zbioru sąsiadów.	Idealny do tworzenia spersonalizowanych rekomendacji, gdzie niektóre aspekty są bardziej istotne niż inne.

Wzór na obliczenie odległości euklidesowej między dwoma punktami P i Q w przestrzeni n-wymiarowej jest następujący:

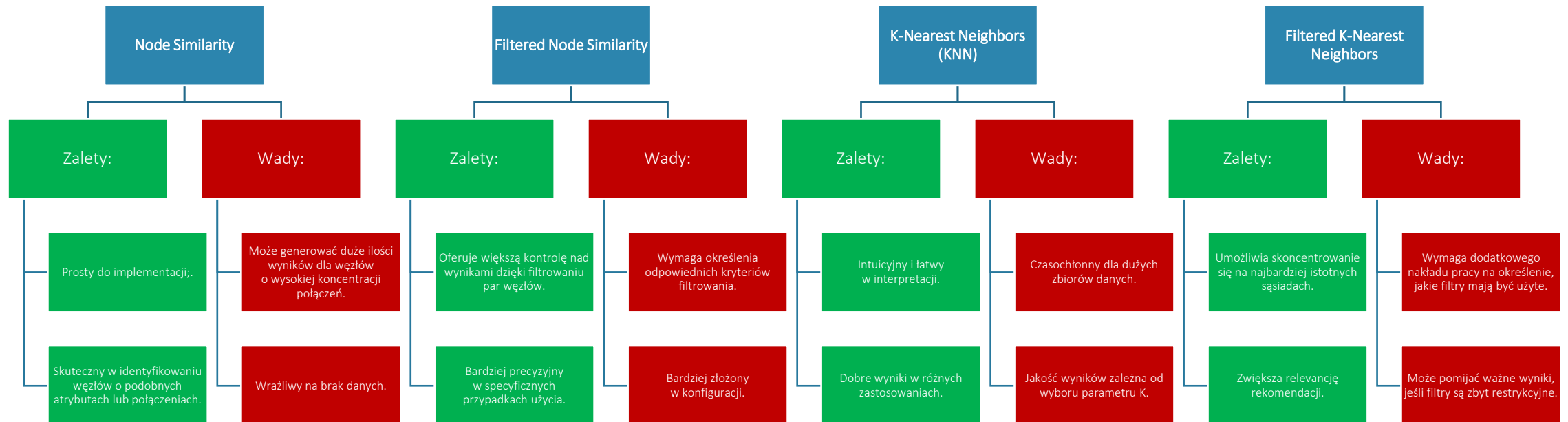
$$d(P, Q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2}$$

gdzie P i Q to punkty o współrzędnych $(p_1, p_2, \dots, p_n)$ i $(q_1, q_2, \dots, q_n)$ odpowiednio. Odległość euklidesowa jest zatem pierwiastkiem kwadratowym sumy kwadratów różnic ich odpowiednich współrzędnych.

Wzór na obliczenie odległości euklidesowej między dwoma punktami A(x_1, y_1) i B(x_2, y_2) w przestrzeni dwuwymiarowej jest następujący:

$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Algorytmy podobieństwa



Algorytmy podobieństwa

Ćwiczenie: Rekomendacja Partnerów Biznesowych w Dynamizującym się Rynku E-commerce

Jako analityk w firmie doradczej stawiasz czoła wyzwaniu dopasowania partnerów biznesowych w szybko zmieniającym się środowisku e-commerce. Twoim zadaniem jest analiza sieci handlowej, która nieustannie ewoluuje ze względu na trendy rynkowe, nowe produkty i zmieniające się preferencje konsumentów. Twoi klienci poszukują firm oferujących produkty, które uzupełnią ich asortyment i otworzą drogę do nowych segmentów rynku lub grup klientów.

Zadanie

Wybierz algorytm, który najlepiej odwzorowuje złożoność sieci handlowych, biorąc pod uwagę następujące kryteria:

- Szybkość adaptacji do zmian w sieci: jak szybko algorytm może aktualizować wyniki podobieństwa w odpowiedzi na nowe dane.
- Skalowalność: zdolność algorytmu do radzenia sobie z dużą liczbą węzłów i krawędzi bez utraty wydajności.
- Precyzja: dokładność algorytmu w identyfikacji najbardziej wartościowych i obiecujących połączeń biznesowych.

Uzasadnij swój wybór.

Algorytmy podobieństwa

Propozycja odpowiedzi

Filtered Node Similarity ten algorytm jest wyposażony w mechanizmy filtrowania, które mogą być szczególnie użyteczne w dynamicznie zmieniającym się środowisku e-commerce, pozwalając na skoncentrowanie się na najbardziej istotnych, aktualnych połączeniach biznesowych. Jego zdolność do szybkiej adaptacji do zmian w sieci i precyzyjnego identyfikowania potencjalnych partnerów czyni go odpowiednim narzędziem do mapowania złożonych relacji między firmami.

Zalety:

- Selektywne wyniki: Filtry pozwalają na ograniczenie wyników do najbardziej relewantnych połączeń, co jest istotne przy dużej ilości danych.
- Dopasowanie do dynamicznych danych: Możliwość szybkiego dostosowania się do zmieniających się warunków rynkowych dzięki aktualizacji filtrów.
- Wysoka precyzja: Dostosowanie kryteriów filtrowania zwiększa szansę znalezienia najbardziej obiecujących połączeń biznesowych.

Wady:

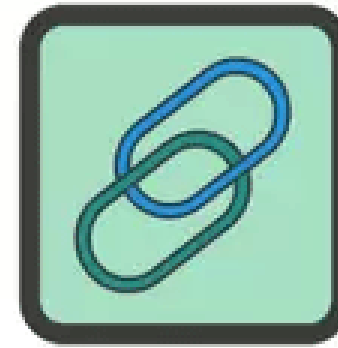
- Złożoność konfiguracji: Ustawienie i zarządzanie filtrami wymaga zaawansowanej wiedzy i zrozumienia kontekstu biznesowego, co może być trudne dla początkujących użytkowników.
- Ryzyko pominięcia danych: Zbyt restrykcyjne filtry mogą skutkować pominięciem ważnych połączeń.

Algorytmy predykcji relacji

Algorytmy predykcji relacji

Na podstawie struktury grafu (relacji) prognozują czy dwa węzły w grafie mogą w przyszłości stworzyć połączenie (krawędź).

- Wspólni sąsiedzi
- Ta sama wspólnota
- Całkowita liczba sąsiadów



- Prognozowanie rozwoju sieci społecznych, np. powstawanie nowych połączeń na platformach społecznościowych.
- Przewidywanie interakcji między białkami w genomice, co może mieć zastosowanie w projektowaniu leków.

Algorytmy predykcji relacji

<https://neo4j.com/docs/graph-data-science/current/algorithms/linkprediction/>

Algorytm	Charakterystyka	Zastosowanie
Adamic Adar	Wskaźnik ten podkreśla wpływ wspólnych sąsiadów o niskich stopniach, sugerując, że małe społeczności są silniejsze.	Wykorzystywany w rekomendacji połączeń w sieciach społecznościowych.
Common Neighbors	Liczba wspólnych sąsiadów dwóch węzłów jest podstawą do oceny prawdopodobieństwa nawiązania połączenia.	Używany do prognozowania powstawania nowych połączeń w sieciach.
Preferential Attachment	Założenie, że węzły o wyższych stopniach są bardziej prawdopodobne do tworzenia nowych połączeń.	Stosowany do analizy wzrostu sieci i mechanizmów tworzenia się połączeń.
Resource Allocation	Sąsiedzi z mniejszymi stopniami mają większy wpływ, odwzorowując realną alokację zasobów w sieciach.	Stosowany w systemach rekomendacji oraz w analizie wydajności sieci.
Same Community	Preferencja do tworzenia połączeń wewnątrz tej samej wspólnoty.	Wykorzystywany do wykrywania i wzmacniania struktury społecznościowej.
Total Neighbors	Ogólna liczba sąsiadów, którą posiadają dwa węzły, używana do oceny ich podobieństwa.	Używany do identyfikacji silnie powiązanych węzłów w sieciach.

Algorytmy predykcji relacji

Adamic Adar

Zalety:

Podkreśla wartość połączeń z rzadkimi wspólnymi sąsiadami, co może sugerować mocniejsze relacje.

Dobra wydajność w przewidywaniu połączeń w sieciach społecznościowych.

Wady:

Może nie być tak efektywny w gęstych sieciach, gdzie wiele węzłów ma podobnie wysokie stopnie.

Może nie uwzględniać pełnego kontekstu połączeń w sieci, ograniczając się do lokalnej topologii.

Common Neighbors

Zalety:

Intuicyjny i prosty do zrozumienia oraz implementacji.

Efektywnie identyfikuje potencjalne połączenia w oparciu o bezpośrednie powiązania.

Wady:

W sieciach o wysokiej gęstości połączeń, liczba wspólnych sąsiadów może być wysoka dla wielu par węzłów.

Nie bierze pod uwagę siły ani jakości połączeń, skupiając się wyłącznie na liczbie połączeń.

Preferential Attachment

Zalety

Jest dobry w modelowaniu sieci, gdzie nowe węzły mają tendencję do łączenia się z bardziej popularnymi węzłami.

Dobrze skaluje się do dużych sieci.

Intuicyjny i łatwy do zrozumienia.

Wady

Nie rozróżnia różnych typów połączeń.

Skłonność do tworzenia "hubów" lub węzłów centralnych.

Może nie być odpowiedni dla sieci, w których nowe węzły nie preferują połączeń z węzłami o wysokim stopniu,

Same Community

Zalety

Jest skuteczny w zachowywaniu oraz identyfikowaniu struktury społeczności.

Dokładnie identyfikuje potencjalne połączenia w sieciach.

Efektywny gdzie społeczności są wyraźnie zdefiniowane i mają silne wewnętrzne połączenia.

Wady

Może nie działać dobrze w sieciach bez wyraźnej struktury społeczności.

Skupia się na lokalnych połączeniach wewnątrz społeczności.

Jest wymagający obliczeniowo dla bardzo dużych sieci.

Algorytmy predykcji relacji

Ćwiczenie: Identyfikacja Kluczowych Połączeń w Sieci

Jesteś analitykiem danych w dużej korporacji. Zostałeś poproszony o opracowanie metody identyfikacji kluczowych połączeń, które mogą odgrywać istotną rolę w przepływie informacji między różnymi działami i zespołami. Organizacja jest dynamiczna i stale się rozwija, dodając nowe zespoły i pracowników. Twoim zadaniem jest zaproponowanie algorytmu, który będzie odporny na zmiany i będzie mógł skalować się wraz ze wzrostem liczby połączeń w organizacji.

Zadanie

1. Wybierz algorytm, który najlepiej radzi sobie z dynamicznym charakterem organizacji i jej skalowalnością.
2. Uzasadnij swój wybór, biorąc pod uwagę specyfikę powiązań wewnątrz organizacji oraz możliwość szybkiego aktualizowania modelu w odpowiedzi na nowe dane.

Algorytmy predykcji relacji

Propozycja odpowiedzi

Algorytm Preferential Attachment może być szczególnie przydatny w tym przypadku, ponieważ opiera się na założeniu, że węzły o wyższym stopniu są bardziej prawdopodobne do tworzenia nowych połączeń. W kontekście dużej, dynamicznej organizacji, ten algorytm może pomóc zidentyfikować kluczowe połączenia, które odgrywają rolę w łączeniu różnych działów i zespołów.

Zalety:

- Jest skuteczny w modelowaniu rzeczywistych sieci, gdzie nowe połączenia są bardziej prawdopodobne między już dobrze połączonymi węzłami. W organizacji, takie podejście może pomóc zidentyfikować pracowników lub zespoły, które odgrywają centralną rolę w komunikacji.
- Algorytm jest dobrze skalowalny do dużych sieci, co jest istotne w przypadku rozrastającej się organizacji.

Wady:

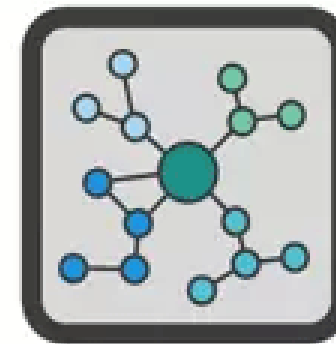
- Skłonność do tworzenia centralnych "hubów" może prowadzić do nadmiernej koncentracji połączeń w kilku kluczowych węzłach, co może zniekształcać obraz całej sieci.
- Algorytm nie uwzględnia różnorodnych typów połączeń, co może być problematyczne w organizacji ze złożoną strukturą połączeń między działami.

Algorytmy centralności

Algorytmy centralności

Pozwalają ocenić, jakie węzły w sieci są najważniejsze, czyli mają mogą mieć największy wpływ.

- Centralność stopnia (Degree Centrality)
- Centralność bliskości (Closeness Centrality)
- Centralność pośrednictwa (Betweenness Centrality)
- Centralność wektora własnego (Eigenvector Centrality)
- PageRank



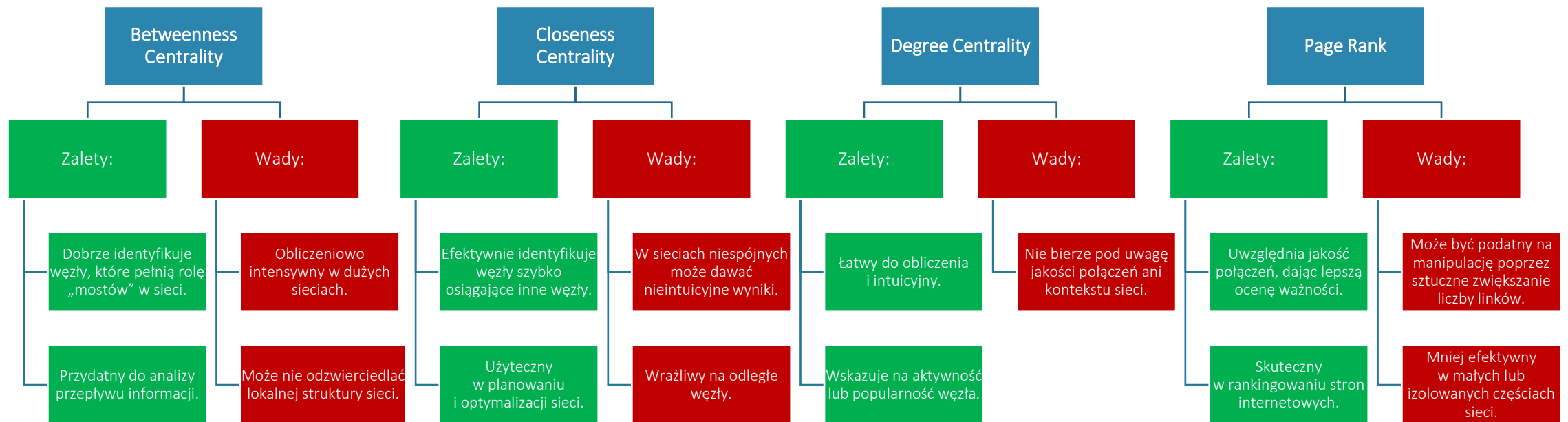
- Analiza struktury organizacji w celu identyfikacji kluczowych pracowników.
- Badanie sieci społecznych, aby znaleźć influencerów lub kluczowe węzły rozprzestrzeniania informacji.

Algorytmy centralności

<https://neo4j.com/docs/graph-data-science/current/algorithms/centrality/>

Algorytm	Charakterystyka	Zastosowanie
Article Rank	Podobny do PageRank, lecz zwraca uwagę na jakość linków wchodzących, nie tylko ich liczbę.	Używany do oceny ważności publikacji i dokumentów.
Betweenness Centrality	Mierzy, ile razy węzeł pojawia się na najkrótszych ścieżkach między parami węzłów.	Stosowany do identyfikowania punktów kontrolnych w sieciach społecznościowych i transportowych.
CELF	Optymalizowany algorytm do wykrywania węzłów z największym potencjałem rozprzestrzeniania informacji.	Efektywny w strategiach marketingowych i analizie sieci społecznościowych.
Closeness Centrality	Mierzy średnią odległość do wszystkich innych węzłów w sieci.	Używany do znajdowania najbardziej wpływowych punktów w sieciach komunikacyjnych.
Degree Centrality	Liczba połączeń węzła z innymi węzłami.	Wykorzystywany do identyfikacji najbardziej aktywnych lub popularnych użytkowników w sieci.
Eigenvector Centrality	Mierzy wpływ węzła na podstawie wpływu sąsiadów.	Znajduje węzły o znaczącym wpływie w sieciach społecznościowych.
Page Rank	Ocena ważności strony internetowej na podstawie liczby i jakości linków do niej.	Stosowany przez wyszukiwarki internetowe do rankingu stron.

Algorytmy centralności

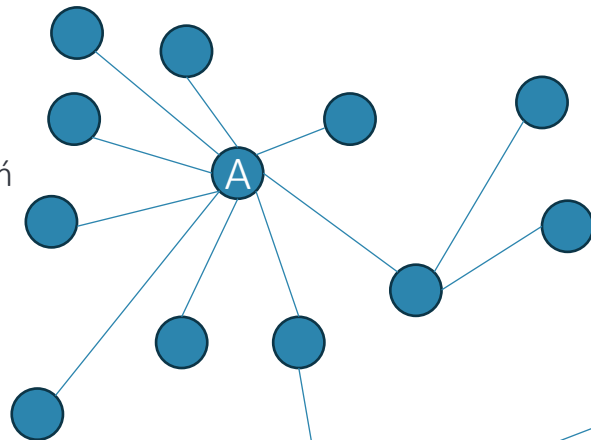


Algorytmy centralności

Degree Centrality
(Centralność stopnia)

Liczba połączeń?

"A" ma najwyższy stopień

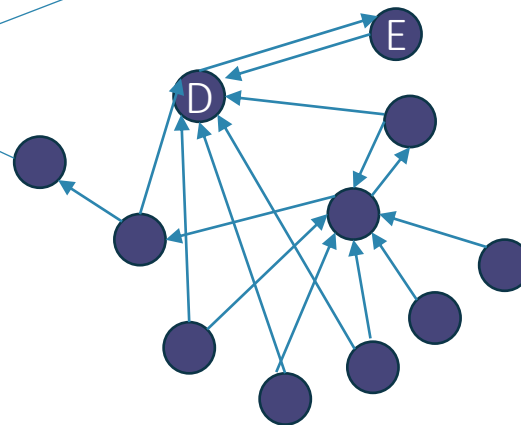


Betweenness Centrality
(Centralność bliskości)

Który węzeł ma największą kontrolę nad przepływem między węzłami i grupami?

"C" jest mostem

C



Louvain Modularity
(Modułowość Louvain)

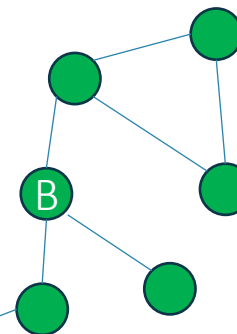
Jak efektywnie podzielić graf na społeczności?



Closeness Centrality
(Centralność pośrednictwa)

Który węzeł może najłatwiej dotrzeć do wszystkich innych węzłów w grafie?

"B" jest najbliższy z najmniejszą liczbą połączeń

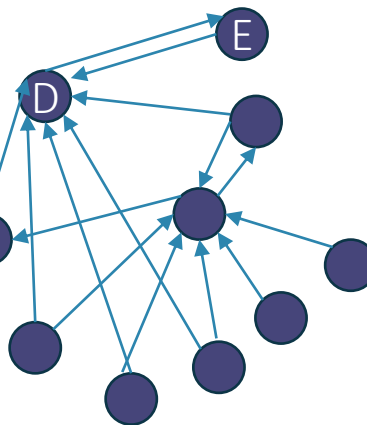


PageRanking

Który węzeł jest najważniejszy?

"D" jest na czele na podstawie liczby i wagi linków przychodzących

"E" jest następny, ze względu na wpływ linku "D"



Algorytmy centralności

Ćwiczenie: Dostosowanie Transportu Publicznego

Wyobraź sobie miasto planujące rozbudowę transportu publicznego. Chcąc maksymalizować efektywność nowych linii, tak aby mieszkańcy mogli jak najszybciej dotrzeć do kluczowych punktów, jak szpitale, szkoły czy centra handlowe. Masz za zadanie analizę istniejącej sieci połączeń, aby zasugerować, gdzie nowe trasy będą miały największy wpływ na poprawę dostępności i efektywności transportu dla mieszkańców. Wybierz odpowiedni algorytm centralności, aby zidentyfikować strategiczne lokalizacje dla przystanków i uzasadnij swój wybór, biorąc pod uwagę zarówno obecne, jak i przewidywane przyszłe zmiany w rozkładzie populacji i użytkowania ziemi.

Zadanie

1. Wybierz algorytm, który najlepiej radzi sobie z dynamicznym charakterem rozwoju miasta.
2. Uzasadnij swój wybór.

Algorytmy centralności

Propozycja odpowiedzi

Algorytm Closeness Centrality pozwala zidentyfikować te węzły (w tym przypadku przystanki lub stacje), które są 'najbliżej' wszystkich innych węzłów w sieci, mierząc średnią odległość do pozostałych punktów. Dla miasta planującego rozbudowę infrastruktury transportowej, pozwoli to na wyznaczenie miejsc, z których mieszkańcy mogą najszybciej dotrzeć do najważniejszych lokalizacji, co jest kluczowe dla efektywnego systemu transportu publicznego. Zapewni to maksymalną dostępność i minimalizację czasu podróży dla największej liczby użytkowników.

Zalety:

- Optymalizacja czasu podróży: Wskazuje lokalizacje, które zmniejszają średni czas dojazdu do kluczowych punktów miasta.
- Efektywna komunikacja: Pomaga w identyfikacji najbardziej centralnych przystanków, przez które przechodzi najwięcej tras.

Wady:

- Może ignorować zagęszczenie: Nie uwzględnia bezpośrednio poziomu ruchu lub liczby pasażerów na danej trasie.
- Wrażliwy na zmiany sieci: Ocenione 'centralne' przystanki mogą stracić na znaczeniu przy przyszłych zmianach w infrastrukturze miejskiej.

Centralność stopnia

(Degree Centrality)

to miara w analizie sieci społecznościowych, która określa, jak wiele połączeń (krawędzi) prowadzi do danego wężła (osoby, wężła sieci).

Definiuje się jako:

$$C_D(v) = \frac{\deg(v)}{n-1}$$

gdzie:

- $\deg(v)$ - oznacza stopień wężła v , czyli liczbę bezpośrednich połączeń (krawędzi) wężła v z innymi wężłami,
- n - to całkowita liczba wężłów w grafie minus 1, ponieważ nie uwzględnia się połączeń z samym sobą.

Wskazuje na to, jak „popularna” jest dana osoba.

Osoby z wysokim stopniem centralności są często uważane za wpływowe lub strategicznie położone w sieci, ponieważ mogą bezpośrednio komunikować się z wieloma innymi osobami lub pełnić rolę ważnego pośrednika w przepływie informacji.

Centralność bliskości

(Closeness centrality)

to miara w analizie sieci społecznościowych, która określa, jak "blisko" węzeł jest do wszystkich innych węzłów w sieci. Jest to odwrotność sumy odległości od danego węzła do wszystkich innych węzłów.

Definiuje się jako:

$$C_C(v) = \frac{1}{\sum_{u \neq v} d(u,v)}$$

gdzie:

- $d(u, v)$ - to odległość między węzłem u a węzłem v , czyli najkrótsza ścieżka między nimi.

Pokazuje jak szybko informacje rozprzestrzeniają się od danej osoby.

Osoby z wysoką centralnością bliskości mogą być kluczowym aktywnym w procesach wymagających szybkiej komunikacji lub dystrybucji zasobów.

Centralność pośrednictwa

(Betweenness centrality)

to miara centralności w grafie, która wskazuje na stopień, w jakim dany węzeł leży na ścieżkach między innymi węzłami.

Definiuje się jako:

$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

gdzie:

- σ_{st} - to liczba najkrótszych ścieżek od węzła s do węzła t ,
- $\sigma_{st}(v)$ - to liczba tych ścieżek, które przechodzą przez węzeł v ,
- suma rozciąga się na wszystkie pary węzłów s i t w sieci.

Wskazuje na to jak ważna jest osoba dla ciągłości kanałów przepływu informacji.

Osoba z wysoką centralnością pośrednictwa działa jako punkt pośredni (HUB) i ma duże znaczenie dla przepływu informacji. Ponad to, ma ona strategiczną rolę, jeśli chodzi o kontrolę nad przepływem informacji lub zasobów.

PageRanking

(PageRank)

to miara wpływu węzła w sieci ze względu na jakość jego połączeń.

Definiuje się jako:

$$P(U) = \frac{1 - d}{N} + d \sum_{V \in B(U)} \frac{P(V)}{L(V)}$$

gdzie:

- d jest współczynnikiem tłumienia, zazwyczaj ustawianym na około 0.85,
- N jest całkowitą węzłów w grafie,
- $B(U)$ jest zbiorem węzłów wskazujących na węzeł U ,
- $P(V)$ jest PageRank węzła V ,
- $L(V)$ jest liczbą krawędzi wychodzących z węzła V .

Pokazuje jak ważna jest dana osoba ze względu na wagę połączeń z innymi ważnymi osobami.

Osoba z wysokim PageRanking jest uważana za szczególnie wpływową. To oznacza, że taka osoba może skutecznie rozpowszechniać informacje i wpływać na opinie w obrębie swojej społeczności.

Algorytm iteracyjny

to proces obliczeniowy, który powtarza zestaw instrukcji (zwany iteracją) do osiągnięcia określonego wyniku lub stanu.

Proces iteracyjny kontynuowany jest do spełnienia warunku zakończenia, takiego jak osiągnięcie zadanej dokładności, wyczerpanie zestawu danych wejściowych lub wykonanie określonej liczby iteracji.

Modułowość Louvain

(Louvain Modularity)

to miara, która ocenia jak dobrze graf dzieli się na moduły, czyli gęsto połączone grupy węzłów z rzadszymi połączeniami pomiędzy grupami.

Definiuje się jako:

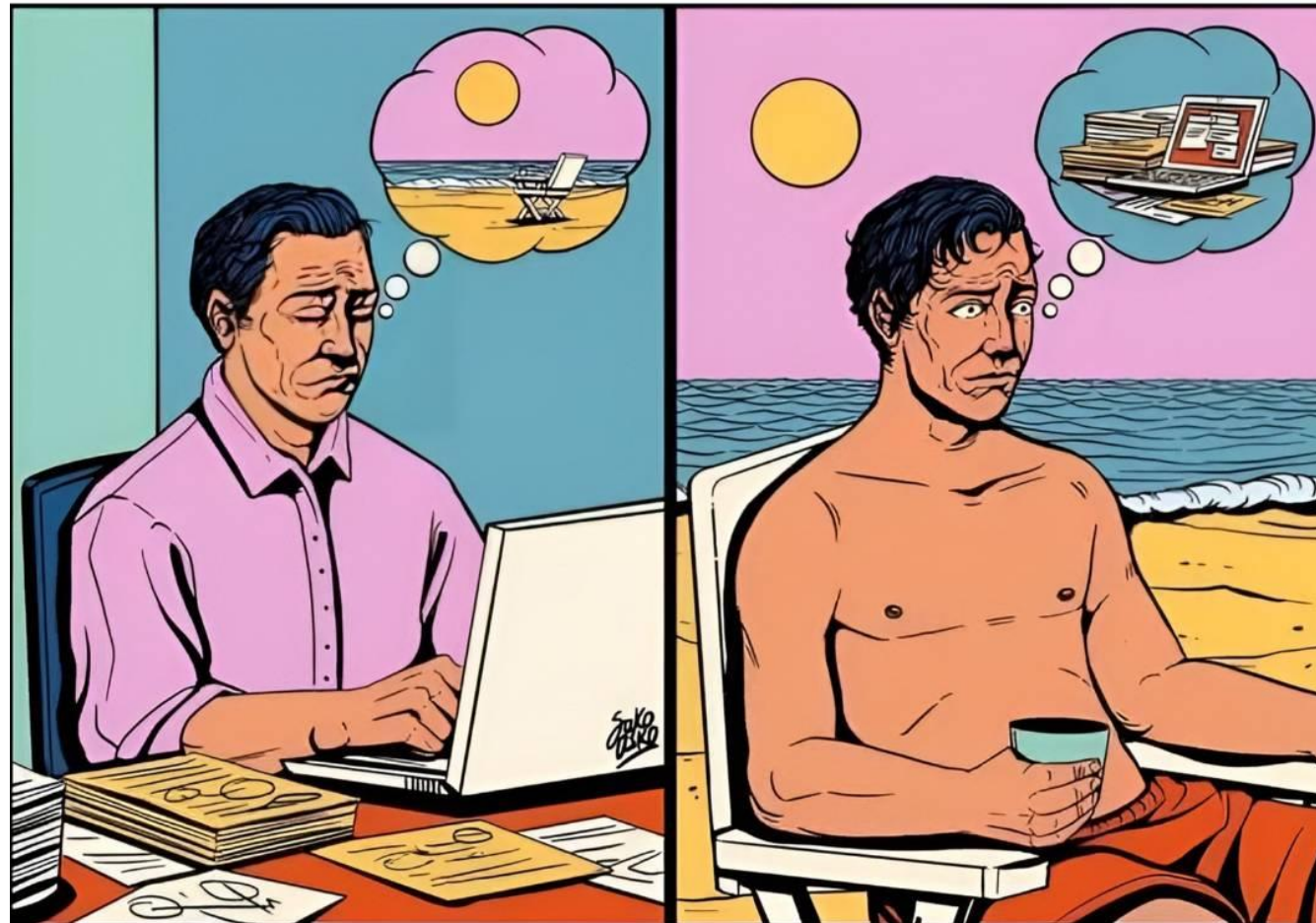
$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

gdzie:

- A_{ij} reprezentuje wagę krawędzi między węzłami i i j ,
- k_i i k_j to sumy wag krawędzi przyłączonych do węzłów i i j ,
- m to suma wszystkich wag krawędzi w sieci,
- c_i i c_j to społeczności, do których należą węzły i i j ,
- δ to funkcja Kroneckera, która jest równa 1, gdy $c_i = c_j$ oraz 0 w przeciwnym przypadku.

Jest stosowany do wykrywania społeczności w dużych sieciach. Identyfikuje grupy o podobnych zainteresowaniach lub wzorcach interakcji.

LIFE OF AN IT PERSON





Analiza grafowa

ANALIZA DANYCH
Z WYKORZYSTANIEM BAZY
GRAFOWEJ Neo4J
I ALGORYTMÓW GRAFOWYCH

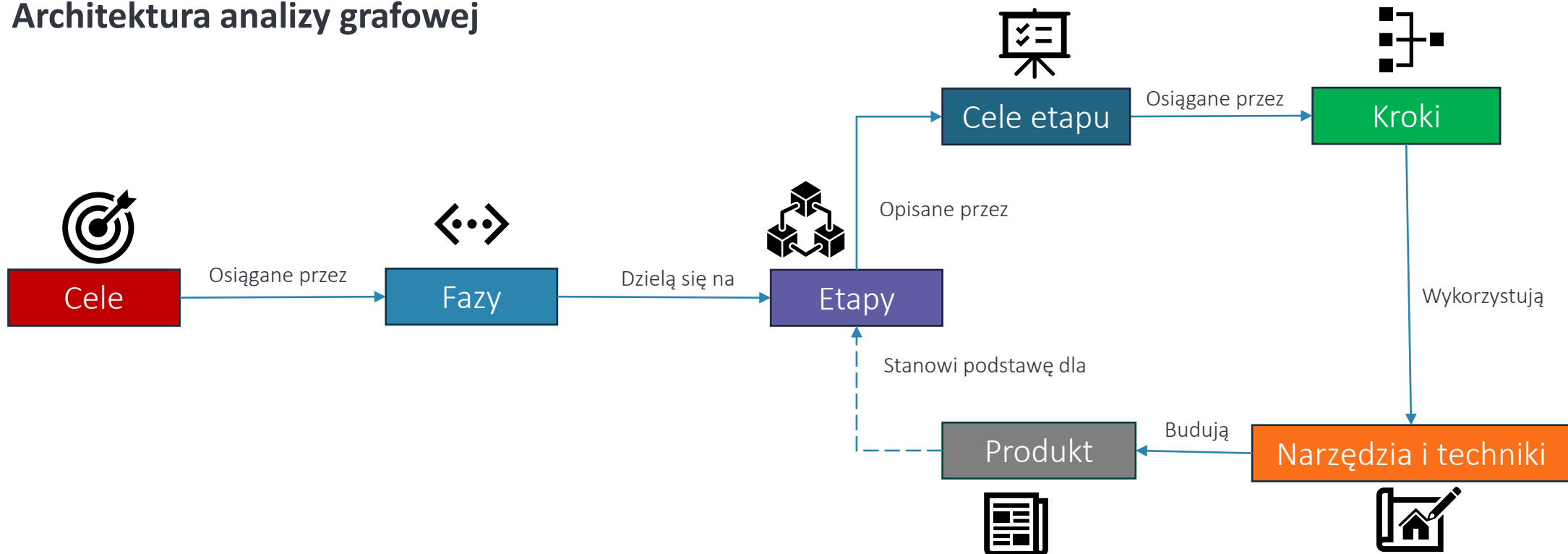


Zagadnienia

- [Architektura analizy grafowej](#)
- [Cele analizy grafowej](#)
- [Fazy analizy grafowej](#)

Architektura analizy grafowej

Architektura analizy grafowej



Architektura analizy grafowej

Analiza grafowa

to metoda badawcza, która wykorzystuje teorię grafów do modelowania, analizowania i interpretowania złożonych zestawów danych reprezentowanych jako węzły (wierzchołki) i krawędzie (relacje).



Cele analizy grafowej

to określone zamierzenia i oczekiwania do obiektu analizy grafowej.



Faza analizy grafowej

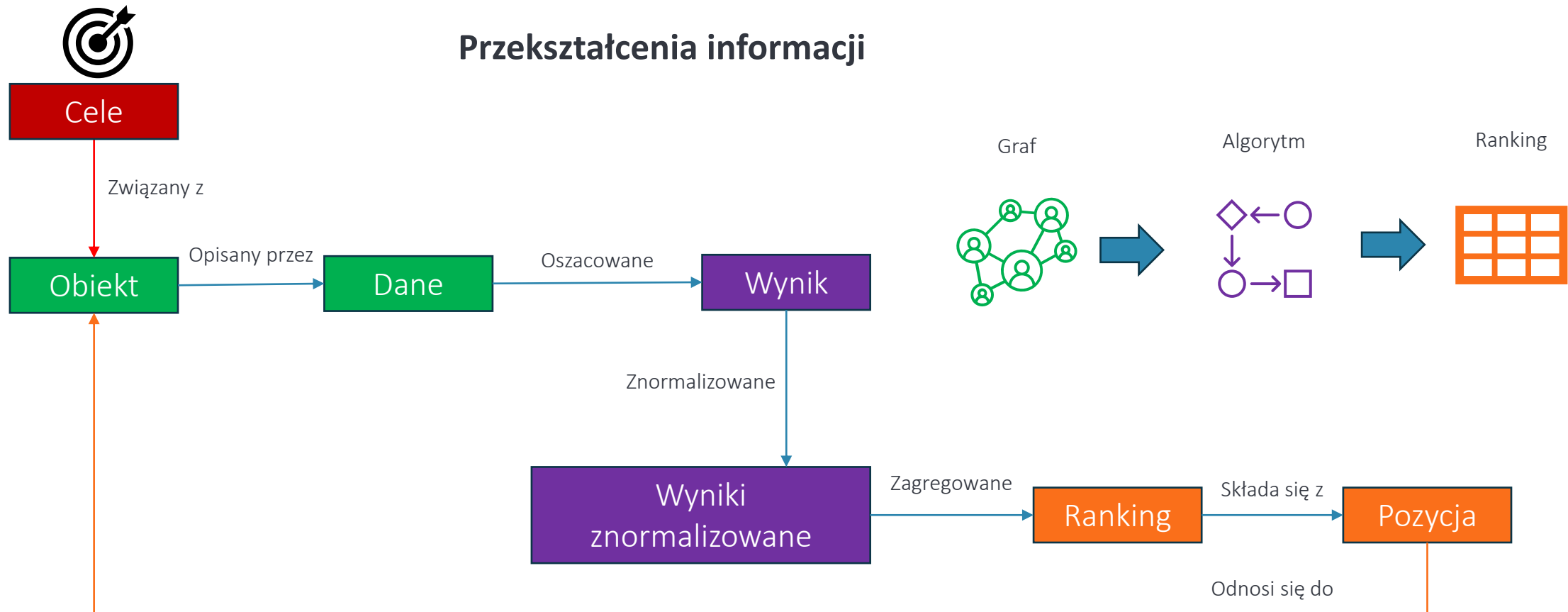
to logicznie spójna sekwencja etapów w ramach analizy grafowej, która dąży do osiągnięcia celów.



Etap analizy grafowej

jest opisany przez cel etapu osiągniany przez zbiór kroków wykorzystujących narzędzia i techniki do budowy produktu. Produkt może stanowić podstawę dla kolejnych etapów.

Architektura analizy grafowej



Architektura analizy grafowej

Obiekt

Reprezentuje element rzeczywistości, który jest elementem celu analizy grafowej. Jest on reprezentowany przez węzeł w bazie grafowej.

Dane

Reprezentują informacje powiązane z obiektem. Są reprezentowane przez krawędzie i właściwości.

Wynik

to rezultat przeprowadzonych obliczeń w ramach działania algorytmu lub innego szacowania na podstawie danych związanych z obiektem.

Wynik znormalizowany

to wynik, który został przeskalowany do zakresu od 0 do 1.

Ranking

przedstawia uporządkowaną listę obiektów powstałą na podstawie zagregowanych znormalizowanych wyników związanych z wagami.

Pozycja (Pozycja w rankingu)

odnosi się do umiejscowienia obiekту w rankingu.

Cele analizy grafowej



Obszar: Ludzie

Pracownicy i zarząd, są podstawowymi elementami każdej organizacji, odpowiedzialni za wykonanie pracy i podejmowanie decyzji.



Badanie relacji

Bazy grafowe mogą wizualizować relacje między pracownikami ujawniając sieć współpracy, linię raportowania i kluczowe węzły komunikacyjne. Dzięki temu zarząd może lepiej rozumieć strukturę zespołów i zoptymalizować przepływ informacji.



Detekcja liderów

Przy pomocy baz grafowych można identyfikować kluczowe osoby w sieciach wewnętrznych, które są krytyczne dla przepływu wiedzy i pracy w organizacji.



Analiza ścieżek kariery

Bazy grafowe mogą być wykorzystywane do analizy i planowania ścieżek kariery pracowników, uwzględniając ich doświadczenie i aspiracje zawodowe.

Cele analizy grafowej



Obszar: Zadania

zadania, procesy i projekty, które organizacja musi wykonać, aby osiągnąć swoje cele.



Planowanie zadań

Bazy grafowe pozwalają na łatwą wizualizację i zrozumienie złożonych zależności między zadaniami, co jest nieocenione przy planowaniu projektów i alokacji zasobów.



Optymalizacja przepływu pracy

Dzięki reprezentacji grafowej można identyfikować wąskie gardła w procesach i szukać możliwości ich optymalizacji, co przekłada się na szybsze i bardziej efektywne realizowanie zadań.



Rozwiązywanie konfliktów

W przypadku konfliktów zasobów lub terminów, bazy grafowe mogą pomóc w ich rozstrzygnięciu poprzez analizę wpływu zmian w jednym obszarze na całą sieć zadań.

Cele analizy grafowej



Obszar: Struktura

sposób, w jaki są organizowane zadania i ludzie, obejmuje hierarchię i podział obowiązków.



Dostrajanie hierarchii

Bazy grafowe umożliwiają intuicyjną wizualizację hierarchicznych struktur organizacyjnych, co może pomóc pracownikom zrozumieć strukturę firmy i ich miejsce w niej.



Analiza sieci organizacyjnych

Umożliwiają analizę sieci formalnych i nieformalnych w organizacji, identyfikując kluczowe węzły (np. osoby wpływowe) i pomagają w zrozumieniu, jak informacje i wpływy przepływają w organizacji.



Przepływ pracy

Umożliwiają modelowanie i analizę przepływu pracy między różnymi działami i zespołami, co może pomóc w identyfikacji wąskich gardeł i potencjalnych ulepszeń.

Cele analizy grafowej



Obszar: Technologia

narzędzia i procesy używane do efektywnego wykonywania zadań, np. oprogramowanie, maszyny, sprzęt komputerowy.



Zależności technologiczne

Bazy grafowe mogą reprezentować złożone relacje między różnymi systemami technologicznymi i ich komponentami. Pozwala to na zrozumienie, jak zmiany w jednym elemencie mogą wpłynąć na inne, co jest niezwykle pomocne w zarządzaniu infrastrukturą IT.



Bezpieczeństwo informacji

Bazy grafowe mogą być wykorzystywane do analizy wzorców w ruchu sieciowym, co pomaga w wykrywaniu niezgodności sugerujących potencjalne zagrożenia bezpieczeństwa.

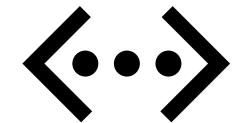


Monitorowanie systemów

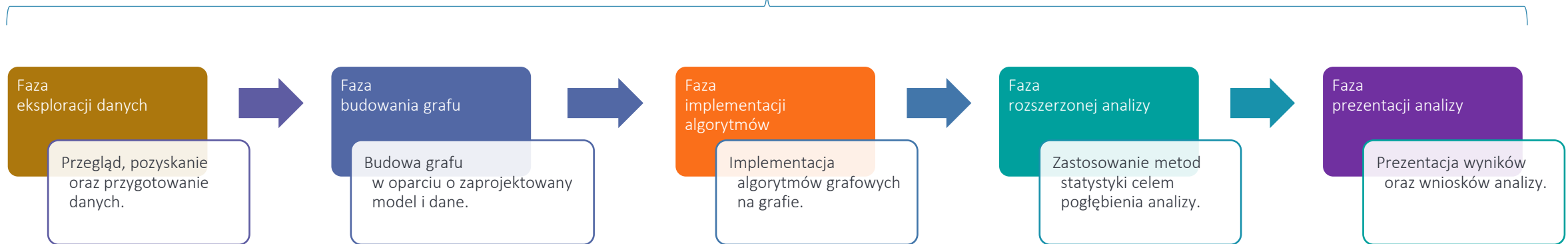
Ułatwiają monitorowanie różnych systemów poprzez zapewnienie jasnego obrazu, jak dane i procesy przepływają między aplikacjami i usługami oraz gdzie są zlokalizowane kluczowe komponenty.

Fazy analizy grafowej

Fazy analizy grafowej

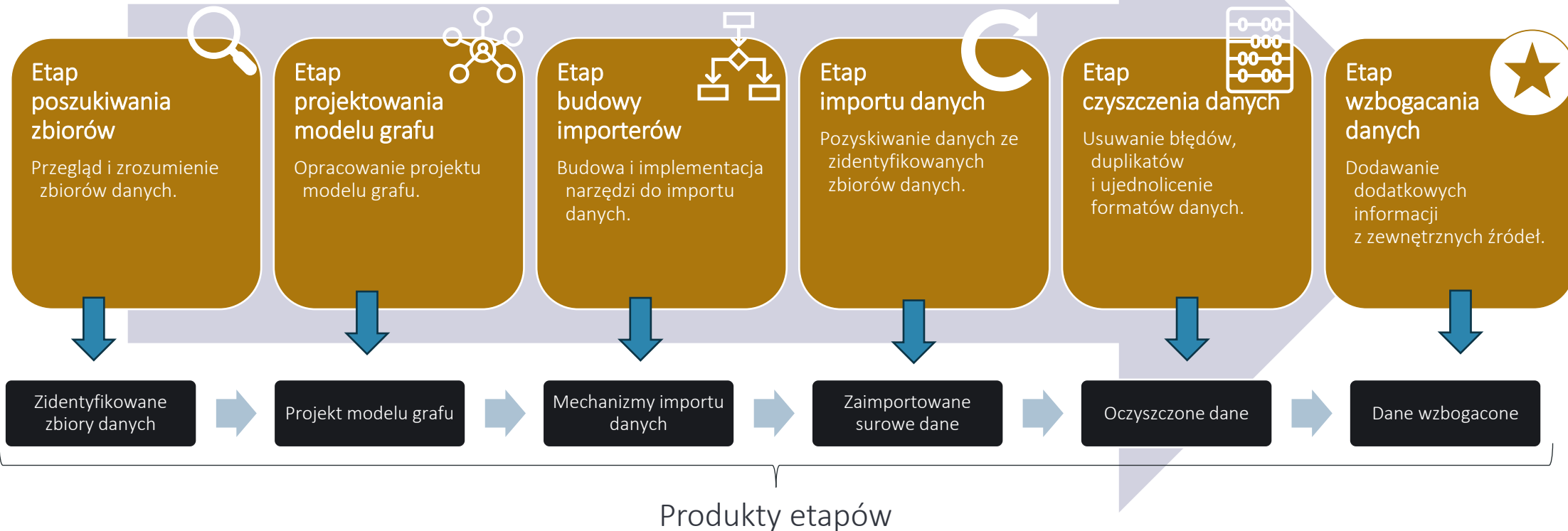


Dążą do osiągnięcia celów analizy grafowej



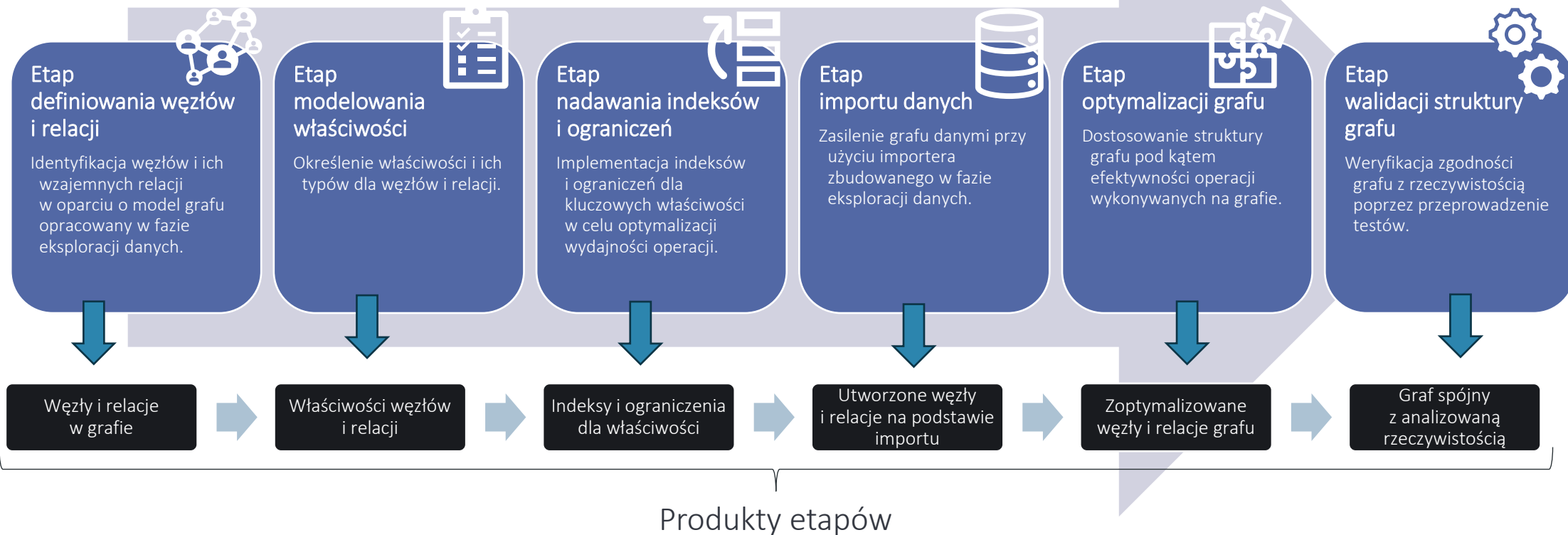
Faza eksploracji danych

Etapy fazy eksploracji danych



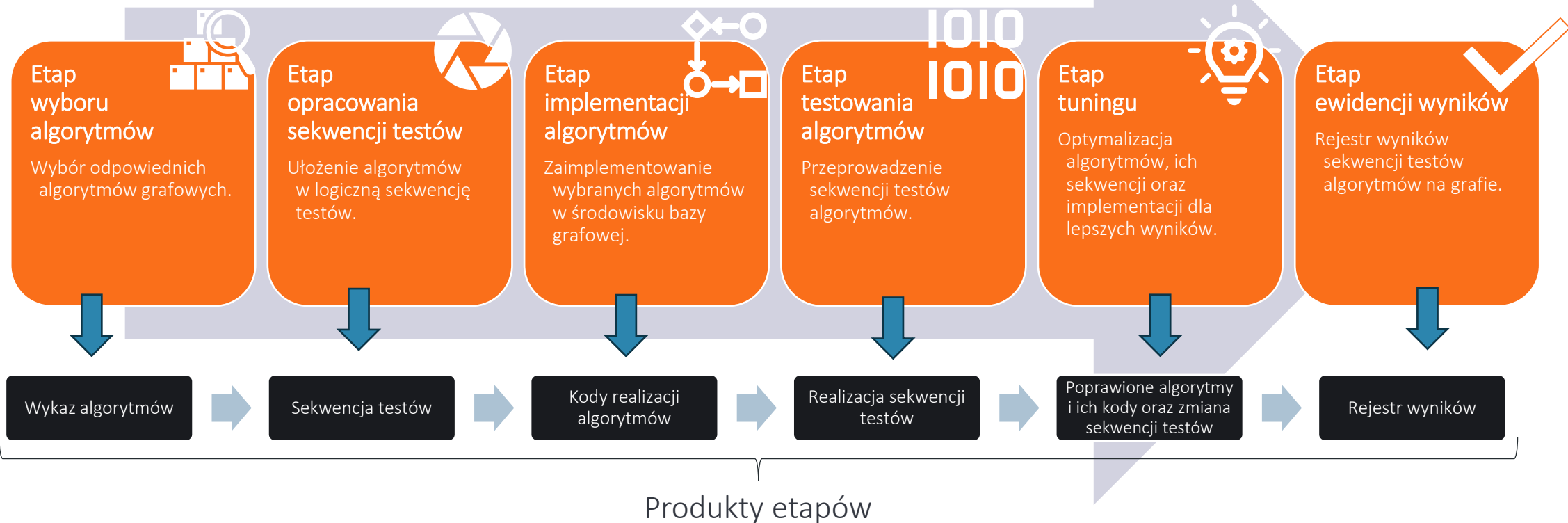
Faza budowy grafu

Etapy fazy budowy grafu



Faza implementacji algorytmów

Etapy fazy implementacji algorytmów



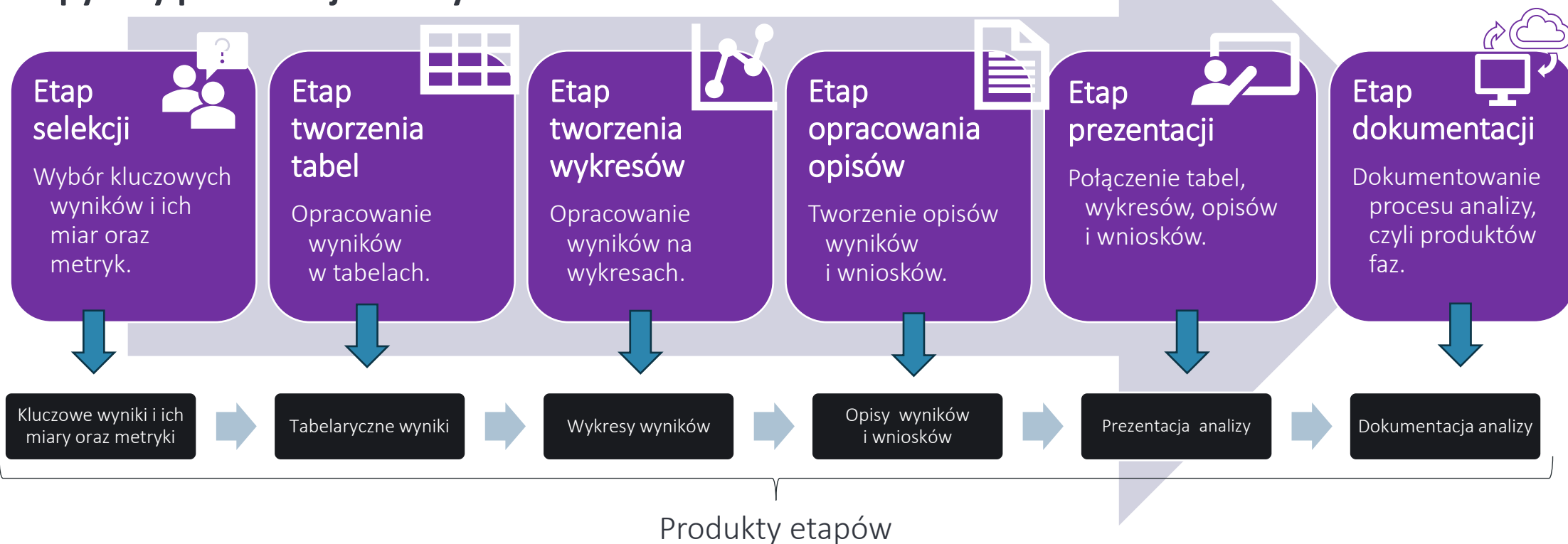
Faza rozszerzonej analizy

Etapy fazy rozszerzonej analizy



Faz prezentacji analizy

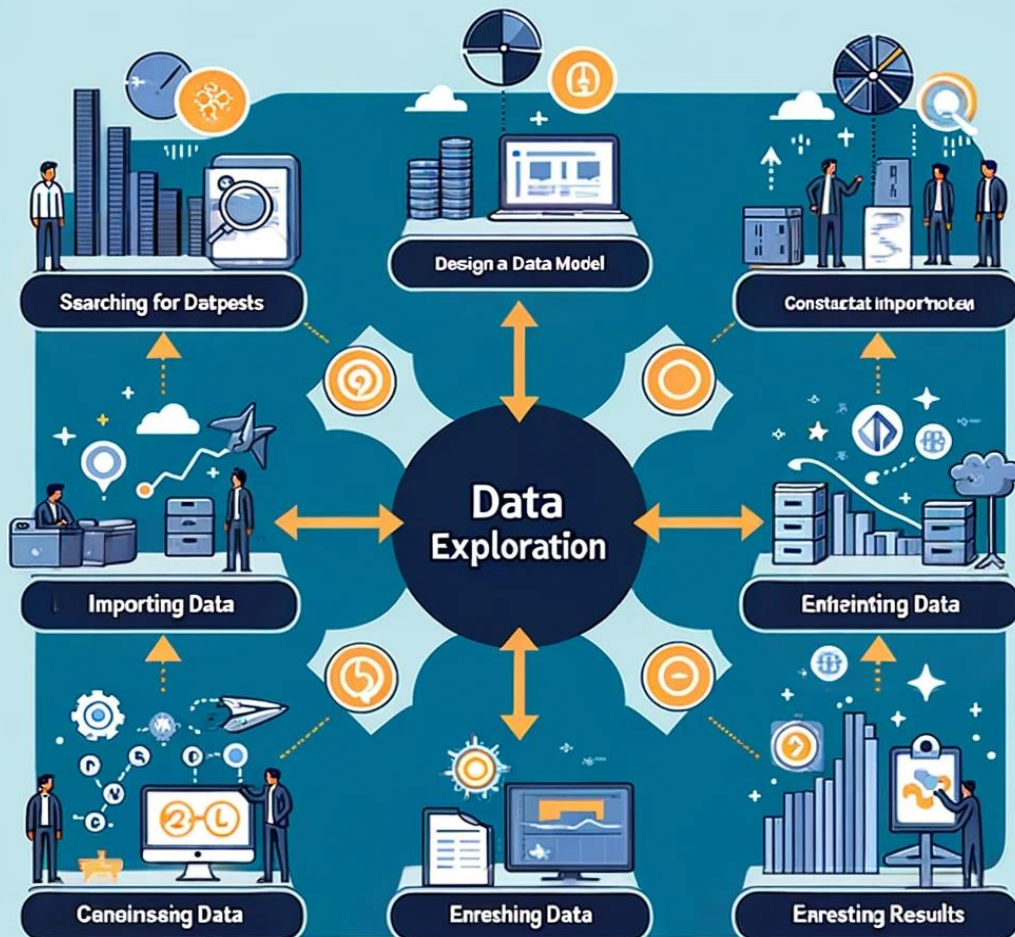
Etapy fazy prezentacji analizy





Faza
eksploracji danych

ANALIZA DANYCH
Z WYKORZYSTANIEM BAZY
GRAFOWEJ Neo4J
I ALGORYTMÓW GRAFOWYCH

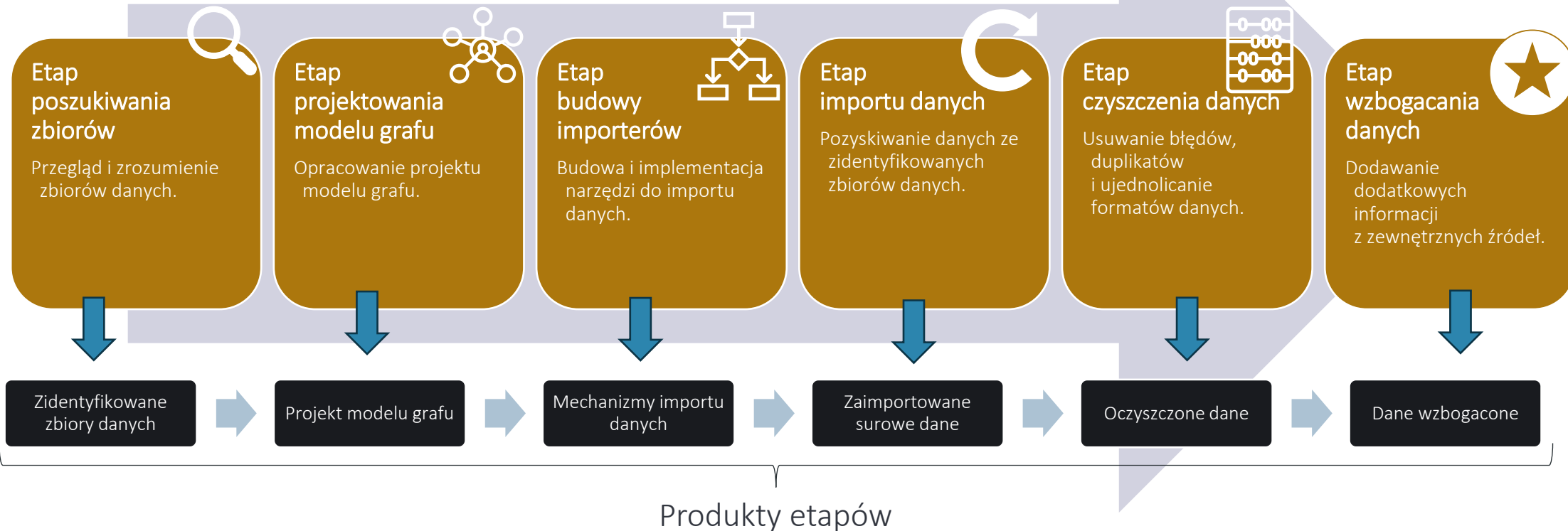


Zagadnienia

- Faza eksploracji danych
 - Etap poszukiwania zbiorów
 - Etap projektu modelu grafu
 - Etap budowy importerów
 - Etap importu danych
 - Etap czyszczenia danych
 - Etap wzbogacenia danych
- Zadanie: Faza eksploracji danych

Faza eksploracji danych

Etapy fazy eksploracji danych



Etap poszukiwania zbiorów

Etap poszukiwania zbiorów

Cel etapu: Identyfikacja i zrozumienie różnorodnych źródeł danych, które będą fundamentem analizy grafowej.

Kroki:

- Wstępny przegląd potencjalnych zbiorów danych.
- Ocena przydatności i wiarygodności danych.
- Selekcja danych na podstawie określonych kryteriów zgodności z celami analizy.

Produkt:

- Zidentyfikowane zbiory danych.

Narzędzia i techniki:

- Wykorzystanie narzędzi do wizualizacji danych w celu ich lepszego zrozumienia.
- Zastosowanie technik analitycznych dla wstępnej oceny jakości danych.



Etap poszukiwania zbiorów

Zbiory danych

Wewnętrzne bazy danych organizacji	Dane z systemów CRM, ERP, czy innych operacyjnych baz danych.
Bazy danych usług informatycznych	Active Directory, poczta elektroniczna, komunikatory, kalendarze, systemy zarządzania projektami.
Systemy monitorowania i logowania	Logi z serwerów, aplikacji, i systemów bezpieczeństwa.
Social media i internet	Dane z mediów społecznościowych, strony internetowe, fora dyskusyjne.
Urządzenia IoT i sensory	Dane generowane przez czujniki i urządzenia połączone z Internetem Rzeczy.
Dane transakcyjne	Historie zakupów, transakcje finansowych, interakcji z serwisami online.
Systemy do zarządzania treścią (CMS)	Treści i metadane z systemów zarządzających treścią.
Dane z aplikacji mobilnych i webowych	Dane użytkowników generowane przez aplikacje.
Publicznie dostępne zbiory danych	Dane rządowe, demograficzne, ekonomiczne, statystyczne udostępniane przez różne organizacje.
Badania rynkowe i ankiety	Dane zebrane z badań, ankiet, ocen i opinii.
Dane zewnętrzne zakupione od dostawców danych	Specjalistyczne zbiory danych dostarczane przez firmy trzecie.



Etap poszukiwania zbiorów

Narzędzia i techniki (Etap poszukiwania zbiorów)



Tableau

Zaawansowane dashboards do wizualizacji, pozwalające na interaktywną eksplorację danych z wewnętrznych baz danych organizacji oraz systemów monitorowania i logowania.

<https://www.tableau.com/products/desktop>

Power BI

Przyjazne dla użytkownika interfejsy do tworzenia wizualizacji danych z baz danych usług informatycznych, urządzeń IoT, czy danych transakcyjnych.

<https://www.microsoft.com/en-us/power-platform/products/power-bi>

Google Data Studio

Narzędzie do wizualizacji danych firmy Google. Pozwala na łączenie danych z różnych źródeł, tworzenie wizualizacji i raportów, a także udostępnianie ich innym użytkownikom.

<https://analytics.google.com/analytics/academy/>

Qlik Sense

Narzędzie do odkrywania danych, szczególnie przydatne w analizie złożonych danych transakcyjnych oraz danych z aplikacji mobilnych i webowych.

<https://www.qlik.com/us/products/qlik-sense>

SQL

Do wyciągania i oczyszczania danych z systemów CMS, baz danych usług informatycznych i wewnętrznych baz danych.

Python

Skrypty umożliwiające szczegółową analizę i wizualizację danych, idealne dla danych z social media i publicznie dostępnych zbiorów danych.

Excel

Uniwersalne narzędzie do podstawowej wizualizacji danych i analizy statystycznej.

Etap projektu modelu grafu

Etap projektu modelu grafu

Cel etapu: Opracowanie projektu modelu grafu, który będzie odwzorowywać dane w postaci węzłów i relacji.

Kroki:

- Definiowanie węzłów, które reprezentować będą obiekty.
- Określanie typów relacji między węzłami, reprezentujących powiązania.
- Projektowanie etykiet oraz właściwości.

Produkt:

- Projekt modelu grafu, gotowy do implementacji w bazie Neo4j (bez implementacji).

Narzędzia i techniki:

- Wykorzystanie narzędzi modelowania grafowego, które mogą wspierać proces projektowania przez graficzną reprezentację struktury grafu.
- Prototypowanie w narzędziach takich jak Neo4j Bloom.



Etap projektu modelu grafu

Narzędzia i techniki (Etap Projektu modelu grafu)



Arrows

Narzędzie do rysowania diagramów grafowych online, które pozwala na szybkie projektowanie i eksport schematów do Cypher.

<https://neo4j.com/labs/arrows/>

Lucidchart

Aplikacja internetowa umożliwiająca tworzenie różnych rodzajów diagramów, w tym grafów, za pomocą przeciągania i upuszczania.

<https://www.lucidchart.com/pages/>

Microsoft Visio

Program do tworzenia diagramów biznesowych, który również może być wykorzystany do modelowania grafów.

Visual Paradigm

Kompleksowe narzędzie do modelowania UML i projektowania baz danych, które również umożliwia modelowanie grafów.

<https://www.visual-paradigm.com/>

Neo4j Bloom

Aplikacja Neo4j, która pozwala na interaktywne eksplorowanie i wizualizację danych grafowych bez pisania zapytań Cypher.

<https://neo4j.com/docs/bloom-user-guide/current/about-bloom/>

Etap budowy importerów

Etap budowy importerów

Cele etapu: Rozwój narzędzi umożliwiających transfer danych do bazy Neo4j, w tym skonfigurowanie procesów ETL, które zapewnią efektywne przenoszenie danych.

Kroki:

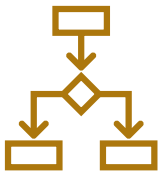
- Projektowanie narzędzi importerów w oparciu o specyfikacje źródeł danych i projekt modelu grafu.
- Implementacja konwersji danych do logiki projektu modelu grafu.
- Testowanie importerów, aby upewnić się, że przekształcanie danych jest poprawne i efektywne.

Produkt:

- Gotowe narzędzia importerów zdolne do przetworzenia i załadowania danych do Neo4j.

Narzędzia i techniki:

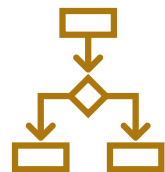
- Skrypty ETL (Extract, Transform, Load) zautomatyzowane procedury wykorzystywane do przetwarzania danych.
- Cypher Query Language w Neo4j do mapowania i przenoszenia danych.
- API języków programowania i konektory baz danych.



Etap budowy importerów

Narzędzia i techniki (Etap budowy importerów)

Neo4j Python Driver	Umożliwia łączenie się z bazą Neo4j i wykonywanie zapytań Cypher bezpośrednio z aplikacji Python.
Neo4j Java Driver	Oferuje obsługę dla aplikacji Java, umożliwiając zarządzanie sesjami, transakcjami i wykonywanie zapytań.
Neo4j JavaScript Driver	Zapewnia interakcję z Neo4j w aplikacjach Node.js oraz w przeglądarkach internetowych.
Cypher's LOAD CSV	Język zapytań Cypher oferuje instrukcję LOAD CSV, która umożliwia bezpośredni import danych z plików CSV.
Neo4j Browser	Graficzny interfejs użytkownika dla Neo4j, który umożliwia importowanie danych z plików CSV i JSON.
Neo4j Shell	Narzędzie wiersza poleceń dołączone do Neo4j, które umożliwia bezpośrednie importowanie danych z plików CSV, JSON i innych formatów.
APOC Library	Rozszerzenie APOC dla Neo4j zawiera zestaw procedur do zaawansowanego importu danych, w tym ładowanie danych z różnych formatów i źródeł.



Etap importu danych

Etap importu danych

Cel etapu: Załadowanie przetworzonych i oczyszczonych danych do bazy Neo4j, zachowując ich strukturę i relacje zgodnie z zaprojektowanym modelem grafu.

Kroki:

- Przeprowadzenie importu.
- Mapowanie i transformacja danych do modelu grafowego.
- Weryfikacja danych w bazie.

Produkt:

- Surowe dane przygotowane do dalszej obróbki.

Narzędzia i techniki:

- Import jednorazowy na żądanie.
- Import cykliczny.



Etap czyszczenia danych

Etap czyszczenia danych

Cel etapu: Usunięcie wszelkich nieścisłości i anomalii z danych, aby zapewnić ich wysoką jakość oraz użyteczność dla analizy grafowej.

Kroki:

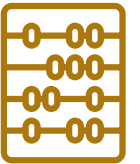
- Identyfikacja i usunięcie duplikatów, błędów i brakujących wartości.
- Normalizacja formatów i typów danych, aby były spójne.
- Weryfikacja i poprawa danych, które nie są zgodne z ustalonymi regułami biznesowymi.
- Modyfikacja importerów.

Produkt:

- Oczyszczone dane i przygotowane do wzbogacenia.

Narzędzia i techniki:

- Wykorzystanie skryptów SQL i Python lub plików CSV i JSON.
- Stosowanie narzędzi ETL do automatyzacji procesów czyszczenia danych.
- Kontrola jakości danych za pomocą modułów weryfikacyjnych.



Etap wzbogacenia danych

Etap wzbogacenia danych

Cel etapu: Wzmocnienie wartości danych przez dodanie nowych informacji, które mogą zwiększyć ich znaczenie dla analizy grafowej oraz doprecyzować wgląd w relacje i wzorce.

Kroki:

- Integracja zewnętrznych źródeł danych, które mogą dodać kontekst lub głębię analizowanym informacjom.
- Wzbogacanie danych o metadane, informacje geograficzne, demograficzne, czy inne dane dodatkowe.
- Uzupełnianie danych o wyniki z badań i analiz, które mogą poprawić ich wartość analityczną.
- Modyfikacja importerów.

Produkt:

- Zestaw danych zintegrowany z dodatkowymi informacjami, gotowy do importu do bazy grafowej.

Narzędzia i techniki:

- API zewnętrznych dostawców danych, by pobrać dodatkowe zestawy danych.
- Skrypty transformacyjne w językach programowania takich jak Python, by zautomatyzować proces wzbogacania.
- Systemy GIS do dodawania danych przestrzennych i geolokalizacyjnych oraz dane GUS (otwarte dane).



Etap wzbogacenia danych

Źródła danych (Etap wzbogacenia danych)



Rejestry publiczne	Takie jak dane rządowe, statystyczne, ekonomiczne, które często są udostępniane przez instytucje publiczne.
Słowniki referencyjne	Zestawy danych, które dostarczają standardowych definicji i identyfikatorów dla różnych obiektów lub pojęć.
Dane geolokalizacyjne	Informacje z systemów GPS i map cyfrowych mogą dodać kontekst przestrzenny.
Open Data Platforms	Platformy takie jak Kaggle czy Google Dataset Search, które udostępniają zbiory danych dla różnych branż.
Dane demograficzne i rynkowe	Raporty i badania rynku dostarczają informacji o trendach i preferencjach konsumentów.
Dane branżowe	Specjalistyczne bazy danych dostępne w konkretnych sektorach, np. w ochronie zdrowia, finansach, czy nieruchomościach.

Przypadek 1.

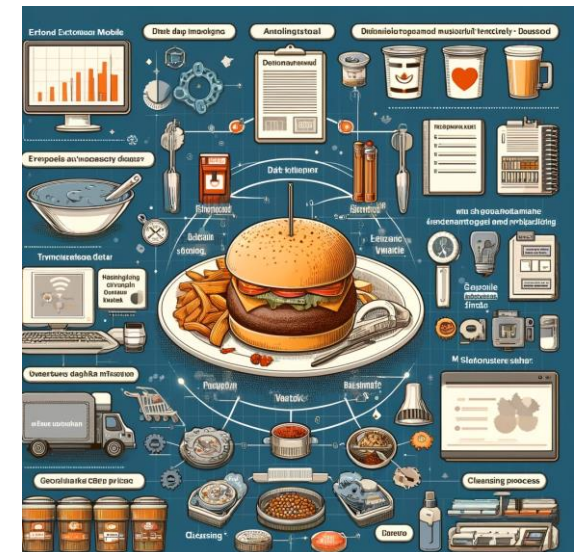
Opracowanie strategii dla sieci restauracji „Ósmy Dzień Tygodnia”

Jesteś analitykiem danych w dynamicznie rozwijającej się sieci restauracji, która dąży do personalizacji swojego menu i optymalizacji procesów zakupowych. Twoje zadanie to wykorzystanie algorytmów grafowych do analizy dużych zbiorów danych, identyfikacji kluczowych trendów konsumenckich oraz optymalizacji operacji wewnętrznych, od menu po łańcuch dostaw.

Kluczowe będzie stworzenie efektywnego procesu importowania, weryfikacji i czyszczenia danych, co pozwoli na szybką reakcję na zmiany na rynku i wewnętrzne potrzeby sieci.

Twoje umiejętności i elastyczne zarządzanie oparte na wiedzy zapewnią restauracji konkurencyjną przewagę na rynku.

Celem jest zaproponowanie na podstawie analizy grafowej nowego Menu dla sieci restauracji „Ósmy Dzień Tygodnia”



Zadanie: Faza eksploracji danych

Dla przypadku 1. w faza eksploracji danych:

1. Zaproponuj źródła danych dla niezbędnych informacji analitycznych.
2. Zaprojektuj model grafu zgodny z celem i pozyskanymi danymi.
3. Zaproponuj narzędzia i techniki do importowania danych ze źródeł.
4. Określ częstotliwość importowania danych (jednorazowo lub okresowo).
5. Zidentyfikuj potencjalne problemy z jakością zaimportowanych danych.
6. Określ, o jakie informacje z zewnętrznych źródeł można wzbogacić dane z systemów restauracji.

Your plan



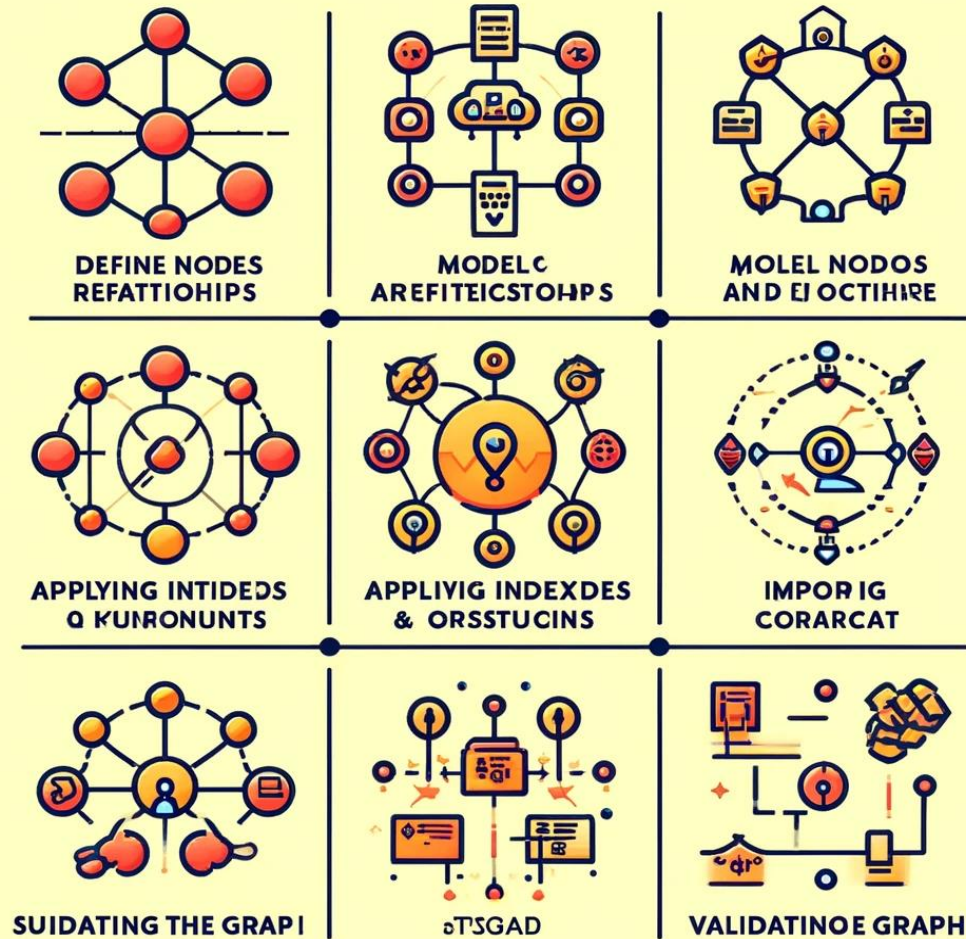
Reality





Faza budowy grafu

ANALIZA DANYCH
Z WYKORZYSTANIEM BAZY
GRAFOWEJ Neo4J
I ALGORYTMÓW GRAFOWYCH

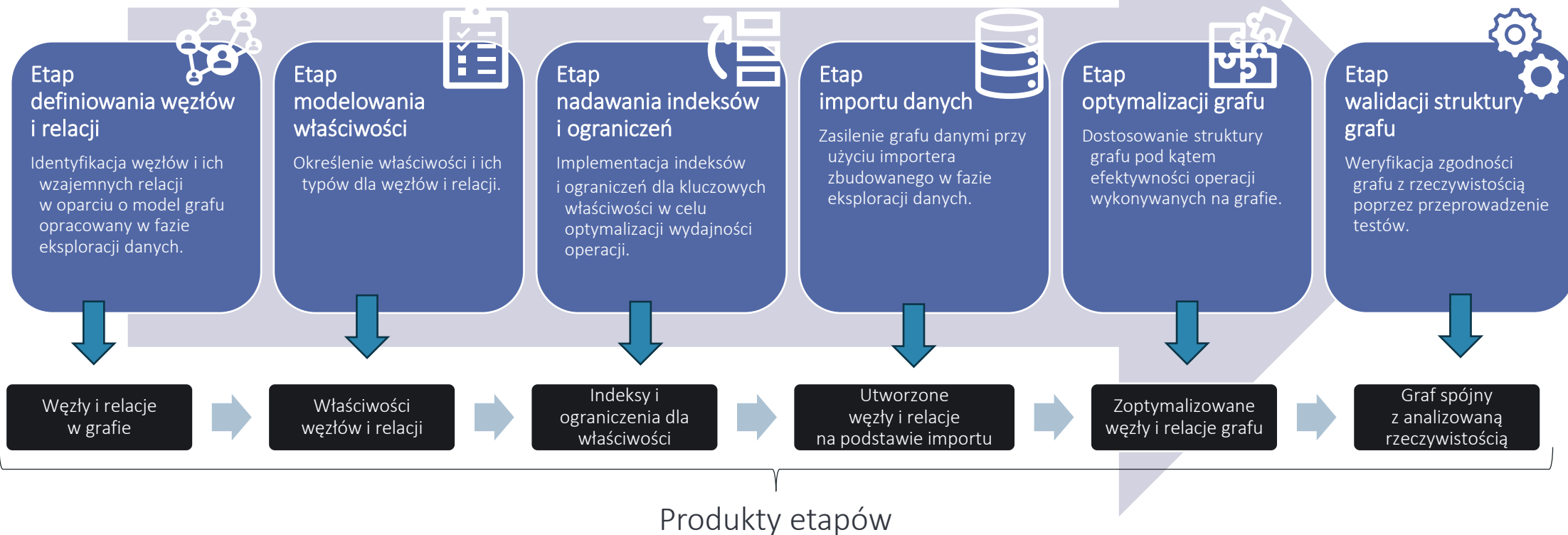


Zagadnienia

- Faza budowy grafu
 - Etap definiowania węzłów i relacji
 - Etap modelowania właściwości
 - Etap nadawania indeksów i ograniczeń
 - Etap importu danych
 - Etap optymalizacji grafu
 - Etap walidacji struktury grafu
- Zadanie: Faza budowy grafu

Faza budowy grafu

Etapy fazy budowy grafu



Etap definiowania węzłów i relacji

Etap definiowania węzłów i relacji

Cel etapu: Budowa głównych elementów grafu (węzłów i krawędzi) w oparciu o model grafu i zestaw danych zintegrowany z dodatkowymi informacjami opracowanych w fazie eksploracji danych.



Kroki:

- Selekcja danych mających stać się węzłami.
- Określenie relacji pomiędzy węzłami.

Produkt:

- Węzły i relacje w grafie.

Narzędzia i techniki:

- Systemy zarządzania bazami danych grafowych, takie jak Neo4j, umożliwiające definiowanie schematu grafu.
- Język zapytań grafowych, np. Cypher w Neo4j, do definiowania struktury grafu.

Etap modelowania właściwości

Etap modelowania właściwości

Cel etapu: Zdefiniowanie i przypisanie właściwości do węzłów i relacji w grafie, które będą używane do analizy.



Kroki:

- Wybór właściwości węzłów i relacji na podstawie cech interesujących w analizie.
- Definiowanie typów danych dla każdej właściwości (np. liczbowe, tekstowe, logiczne).
- Określenie metod obliczeniowych i agregacji dla właściwości, które będą przetwarzane.

Produkt:

- Właściwości węzłów i relacji.

Narzędzia i techniki:

- Systemy zarządzania bazami danych grafowych, takie jak Neo4j, umożliwiające definiowanie schematu grafu.
- Język zapytań grafowych, np. Cypher w Neo4j, do definiowania struktury grafu.

Etap nadawania indeksów i ograniczeń

Etap nadawania indeksów i ograniczeń

Cel etapu: Zwiększenie wydajności zapytań i zapewnienie integralności danych poprzez odpowiednie użycie indeksów i ustanowienie ograniczeń na właściwościach węzłów i relacji.



Kroki:

- Identyfikacja właściwości węzłów i relacji, które często uczestniczą w zapytaniach, w celu stworzenia dla nich indeksów.
- Określenie i ustanowienie ograniczeń unikalności, wymagalności oraz innych reguł biznesowych.
- Konfiguracja indeksów pełnotekstowych, jeśli potrzebne jest wyszukiwanie pełnotekstowe.
- Testowanie wydajności zapytań z indeksami i bez nich, aby ocenić skuteczność indeksowania.

Produkt:

- Indeksy i ograniczenia dla właściwości.

Narzędzia i techniki:

- Systemy zarządzania bazami danych grafowych, takie jak Neo4j, umożliwiające definiowanie schematu grafu.
- Język zapytań grafowych, np. Cypher w Neo4j, do definiowania struktury grafu.
- Narzędzia do testowania wydajności, takie jak profilery zapytań, które pomagają w ocenie wpływu indeksów na czas wykonania zapytań.

Etap importu danych

Etap importu danych

Cel etapu: Wprowadzenie zebranych danych do bazy grafowej w postaci węzłów i relacji, tak aby były one gotowe do analizy.



Kroki:

- Ustalenie częstotliwości importu danych (jednorazowo lub cyklicznie).
- Ustalenie okresu z którego są importowane dane
- Konfiguracja narzędzi automatyzujących zadanie importu danych
- Zasilenie grafu danymi przy użyciu importera zbudowanego w fazie eksploracji danych.
- Weryfikacja poprawności danych po imporcie, sprawdzenie czy dane zostały prawidłowo zmapowane i zaimportowane.

Produkt:

- Utworzone węzły i relacje na podstawie importu.

Narzędzia i techniki:

- Importer zbudowany w fazie eksploracji danych.
- Narzędzia do automatyzacji zadań w celu wprowadzenia cykliczności lub importów uruchamianych jako zdarzenia.

Etap optymalizacji grafu

Etap optymalizacji grafu

Cel etapu: Udoskonalenie struktury i wydajności grafu, aby maksymalizować skuteczność analizy i zapytań.



Kroki:

- Analiza wydajności obecnej struktury grafu z wykorzystaniem typowych zapytań i operacji.
- Reorganizacja grafu, która może obejmować refaktoryzację schematu, usuwanie niepotrzebnych węzłów lub relacji lub ich edycję.
- Przeprowadzenie testów wydajności, aby ocenić wpływ wprowadzonych zmian.

Produkt:

- Zoptymalizowane węzły i relacje grafu.

Narzędzia i techniki:

- Narzędzia do profilowania wydajności, takie jak Neo4j Profiler, umożliwiające dokładne śledzenie i analizę czasów odpowiedzi.
- Systemy monitorowania wydajności, które zapewniają ciągłe śledzenie wydajności i umożliwiają szybką interwencję w razie potrzeby.

Etap walidacji struktury grafu

Etap walidacji struktury grafu

Cel etapu: Zapewnienie, że zbudowany graf poprawnie odwzorowuje obiekty (węzły) i zależności (relacje) występujące w rzeczywistych danych, a także sprawdzenie integralności i spójności struktury.

Kroki:

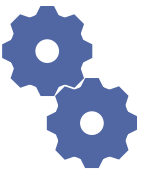
- Recenzja struktury grafu zgodnie z zaprojektowanym modelem, aby upewnić się, że wszystkie węzły i krawędzie są poprawnie zdefiniowane i skonfigurowane.
- Użycie zapytań testowych do symulacji, monitorując zachowanie grafu i jego wydajność.
- Identyfikacja i naprawa niezgodności oraz błędów danych, takich jak osierocone węzły, nieprawidłowe relacje czy niespójne właściwości.
- Przegląd zasad walidacji i ograniczeń w grafowej bazie danych, aby upewnić się, że są one właściwie wdrożone i funkcjonują zgodnie z oczekiwaniami.

Produkt:

- Graf spójny z analizowaną rzeczywistością.

Narzędzia i techniki:

- Systemy zarządzania bazami danych grafowych, takie jak Neo4j, umożliwiające definiowanie schematu grafu.
- Język zapytań grafowych, np. Cypher w Neo4j, do definiowania struktury grafu.
- Narzędzia do testowania wydajności, takie jak profilery zapytań, które pomagają w ocenie wpływu indeksów na czas wykonania zapytań.



Zadanie: Faza budowy grafu

Dla przypadku 1. w fazie budowy grafu:

1. Zidentyfikuj węzły i relacje dla reprezentacji w grafie.
2. Przypisz właściwości do węzłów i relacji dla analiz.
3. Zdefiniuj indeksy dla kluczowych właściwości w celu optymalizacji zapytań.
4. Ustal ograniczenia na właściwościach dla zapewnienia integralności danych.
5. Przedstaw propozycję automatyzacji importu danych do grafu.
6. Jak zweryfikować adekwatność struktury grafu do rzeczywistych danych?



RICH

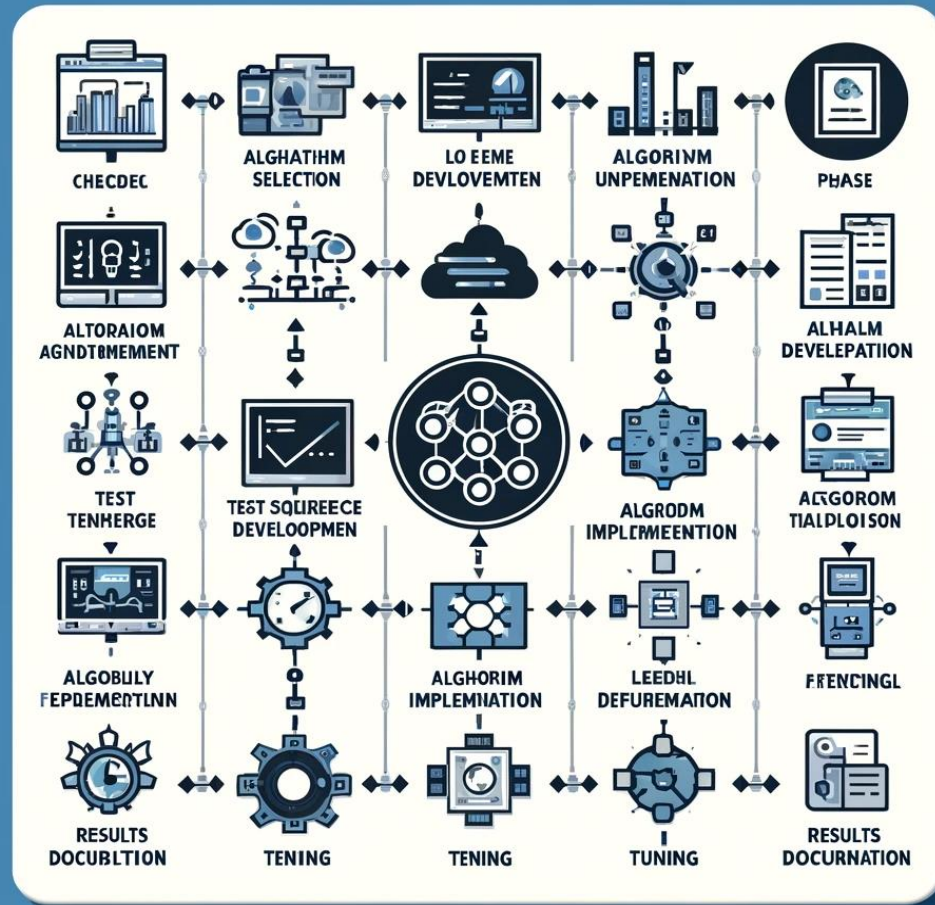


WEALTHY



Faza implementacji algorytmów

ANALIZA DANYCH
Z WYKORZYSTANIEM BAZY
GRAFOWEJ Neo4J
I ALGORYTMÓW GRAFOWYCH

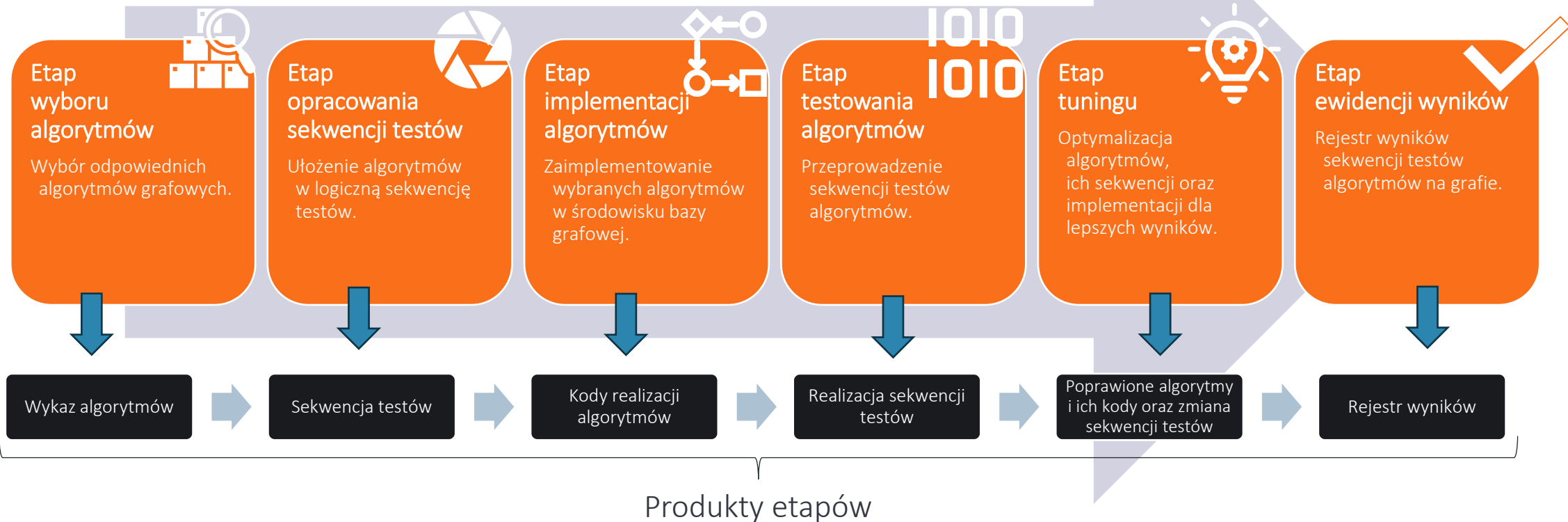


Zagadnienia

- Faza implementacji algorytmów
 - Etap wyboru algorytmów
 - Etap opracowania sekwencji testów
 - Etap implementacji algorytmów
 - Etap testowania algorytmów
 - Etap tuningu
 - Etap ewidencji wyników
- Zadanie: Faza implementacji algorytmów

Faza implementacji algorytmów

Etapy fazy implementacji algorytmów



Etap wyboru algorytmów

Etap wyboru algorytmów

Cel etapu: Wybór odpowiednich algorytmów grafowych, które będą skutecznie realizować określone cele analizy w kontekście dostępnych danych.



Kroki:

- Analiza wymagań poprzez zrozumienie oraz specyfikacja celów analizy grafowej.
- Przegląd dostępnych algorytmów grafowych i ocena ich efektywności, złożoności oraz przydatności.
- Wybór najbardziej odpowiednich algorytmów bazując na przeprowadzonej analizie i ich wydajności.

Produkt:

- Wykaz algorytmów.

Narzędzia i techniki:

- Język zapytań grafowych, np. Cypher w Neo4j.
- Dedykowane biblioteki z algorytmami grafowymi.
- Zasięgnięcie opinii specjalistów ds. algorytmów grafowych, aby uzyskać wgląd w najlepsze praktyki i nowoczesne podejścia w ich doborze.

Etap opracowania sekwencji testów

Etap opracowania sekwencji testów

Cel etapu: Ułożenie algorytmów w logiczną sekwencję testów, aby ich wyniki uzupełniały się.

Kroki:

- Przygotowanie sekwencji testów poprzez zdefiniowanie scenariusza ich realizacji.
- Analiza możliwych zakresów wyników dla poszczególnych algorytmów.
- Powiązanie wyników dla testów i ich możliwa interpretacja.

Produkt:

- Sekwencja testów.

Narzędzia i techniki:

- Język zapytań grafowych, np. Cypher w Neo4j.
- Dedykowane biblioteki z algorytmami grafowymi.
- Zasięgnięcie opinii specjalistów ds. algorytmów grafowych, aby uzyskać wgląd w najlepsze praktyki i nowoczesne podejścia w ich doborze.



Etap implementacji algorytmów

Etap implementacji algorytmów

Cel etapu: Zaimplementowanie sekwencji algorytmów grafowych w środowisku bazy grafowej.

Kroki:

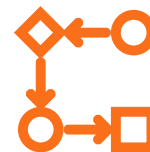
- Przygotowanie środowiska programistycznego, w tym konfiguracja narzędzi i dostęp do baz danych.
- Implementacja algorytmów grafowych lub ich wywołań zgodnie z ustalonymi specyfikacjami.
- Zapewnienie mechanizmów gromadzenia wyników dla poszczególnych testów.

Produkt:

- Kody realizacji algorytmów.

Narzędzia i techniki:

- Język zapytań grafowych, np. Cypher w Neo4j, do implementacji algorytmów grafowych.
- Środowiska programistyczne i narzędzia do programowania, takie jak IDEs, które wspomagają rozwój kodu.
- Biblioteki i frameworki specjalistyczne dla grafów, które ułatwiają implementację algorytmów grafowych.



Etap testowania algorytmów

Etap testowania algorytmów

Cel etapu: Przeprowadzenie testów (uruchomienia) zaimplementowanych algorytmach grafowych zgodnie z sekwencją.

Kroki:

- Wykonanie sekwencji testów (uruchomienia) sekwencji algorytmów grafowych.
- Ewidencja wyników testów.
- Identyfikacja wszelkich błędów lub nieoczekiwanych zachowań algorytmów.

Produkt:

- Realizacji sekwencji testów.

Narzędzia i techniki:

- Język zapytań grafowych, np. Cypher w Neo4j, do implementacji algorytmów grafowych.
- Narzędzia do automatyzacji testów, które powtarzają przeprowadzanie testów i ewidencje wyników.
- Biblioteki i frameworki specjalistyczne dla grafów, które ułatwiają implementację algorytmów grafowych.



Etap tuningu

Etap tuningu

Cel etapu: Poprawa wydajności zaimplementowanych algorytmów grafowych w celu poprawy ich wyników i szybkości działania.



Kroki:

- Identyfikacja wszelkich błędów lub nieoczekiwanych zachowań algorytmów.
- Dostosowanie parametrów algorytmów, aby lepiej pasowały do specyficznych wymagań danych i zapytań.
- Iteracyjne dostosowywanie i testowanie, aż do osiągnięcia akceptowalnych wyników.

Produkt:

- Poprawione algorytmy i ich kody oraz zmiana sekwencji testów.

Narzędzia i techniki:

- Język zapytań grafowych, np. Cypher w Neo4j, do implementacji algorytmów grafowych.
- Biblioteki optymalizacyjne i środowiska programistyczne umożliwiające eksperymentowanie z różnymi konfiguracjami algorytmów.
- Konsultacje z ekspertami w dziedzinie algorytmów grafowych i optymalizacji, aby uzyskać wskazówki dotyczące najlepszych praktyk w tuningu algorytmów.

Etap ewidencji wyników

Etap ewidencji wyników

Cel etapu: Dokumentacja i rejestracja wyników testów ostatecznych wersji algorytmów grafowych umożliwiającą dalszą analizę.

Kroki:

- Zbieranie wyników z przeprowadzonych testów (uruchomień) sekwencji algorytmów.
- Archiwizacja wyników do dalszego wykorzystania w ramach analizy.

Produkt:

- Rejestr wyników.

Narzędzia i techniki:

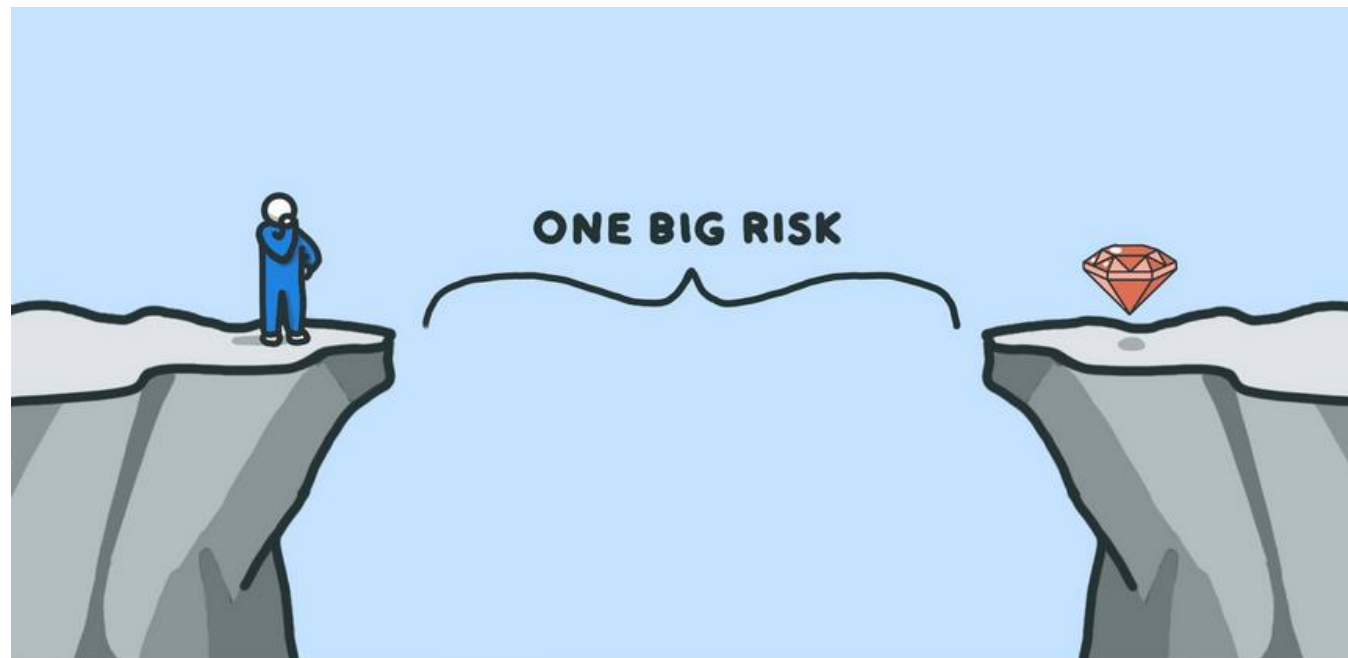
- Arkusze kalkulacyjne (np. Microsoft Excel, Google Sheets).
- Relacyjne bazy danych (np. MySQL, PostgreSQL, MongoDB).
- Systemy do analizy danych (np. R, Python z pakietami pandas i scikit-learn, Tableau, Power BI).



Zadanie: Faza implementacji algorytmów

Dla przypadku 1. w fazie implementacji algorytmów:

1. Dobierz algorytmy grafowe do celu i danych.
2. Zaproponuj sekwencję testów w logicznej kolejności.
3. Opisz wyniki algorytmów w sekwencji.
4. Określ algorytmy o potencjalnych problemach wydajnościowych.
5. Wskaż, czy zmiana parametrów algorytmów jest konieczna i jakie parametry mogą wymagać modyfikacji.
6. Zaproponuj technikę i formę zbierania wyników działania algorytmów.





Faza rozszerzonej analizy

ANALIZA DANYCH
Z WYKORZYSTANIEM BAZY
GRAFOWEJ Neo4J
I ALGORYTMÓW GRAFOWYCH



Zagadnienia

- Faza rozszerzonej analizy
 - Etap korelacji
 - Etap outsiderów
 - Etap normalizacji
 - Etap rankingu
 - Etap centralności
 - Etap rozproszenia
- Zadanie: Faza rozszerzonej analizy

Faza rozszerzonej analizy

Etap fazy rozszerzonej analizy



Etap outsiderów

Etap outsiderów

Cel etapu: Identyfikacja i analiza wartości odstających (outliers), które mogą znacząco wpływać na wyniki analizy w celu wykrycia danych, które odbiegają od normy, co może wskazywać na błędy w danych lub nietypowe przypadki, które wymagają szczególnej uwagi.



Kroki:

- Przegląd danych w poszukiwaniu nietypowych lub skrajnych wartości, które mogą wypaczać ogólne wyniki.
- Użycie technik statystycznych, takich jak metoda IQR (międzykwartyłowy zakres rozstępu) lub z-scores, aby matematycznie zidentyfikować wartości odstające.
- Ocena, jak identyfikowane wartości odstające wpływają na analizę i wyniki algorytmów grafowych, a także decyzja o ich ewentualnym usunięciu lub zachowaniu w analizie.

Produkt:

- Wykaz outlierów.

Narzędzia i techniki:

- Oprogramowanie statystyczne, jak R czy Python z bibliotekami do analizy danych, które umożliwiają efektywne przeprowadzenie obliczeń.
- Skrypty automatyzujące proces, co pozwala na szybkie ponowne analizy w miarę dostarczania nowych danych testowych.
- Arkusze kalkulacyjne: Microsoft Excel, Google Sheets.

Etap outsiderów

Metoda IQR (Interquartile Range)

Kiedy stosować:

- Gdy dane zawierają skrajne wartości.
- Gdy trzeba zdefiniować granice dla akceptowalnych wartości dla danych.

Zalety:

- Odporność na wartości odstające.
- Uniwersalność zastosowania.

Wzór:

$$IQR = Q3 - Q1$$

gdzie:

$Q1$ - to pierwszy kwartyl (25. percentyl), czyli wartość poniżej której znajduje się 25% danych X

$Q3$ - to trzeci kwartyl (75. percentyl), czyli wartość poniżej której znajduje się 75% danych X

Wynik:

$X_{min} = Q1 - 1.5 \times IQR$ - dolna granica odstających wartości

$X_{max} = Q3 + 1.5 \times IQR$ - górna granica odstających wartości

$$X' = \{x_i : x_i < X_{min} \wedge x_i > X_{max}\}$$

Etap outsiderów

Metoda IQR (Interquartile Range)

Przykład: Analiza odwiedzin stron sklepu internetowego.

Numer strony i	Liczba odwiedzin X_i	Dolna granica X_{min}	Górna granica X_{max}	$X < X_{min}$	$X > X_{max}$
1	10	-373,13	831,9		
2	180	-373,13	831,9		
3	160	-373,13	831,9		
4	1045	-373,13	831,9		1
5	15	-373,13	831,9		
6	100	-373,13	831,9		
7	155	-373,13	831,9		
8	10	-373,13	831,9		
9	162	-373,13	831,9		
10	1000	-373,13	831,9		1
11	150	-373,13	831,9		
12	980	-373,13	831,9		1

Q1	78,75
Q2	157,50
Q3	380,00
Q4	1045,00
IQR	301,25

Wynik:

$X' = \{x_4, x_{10}, x_{12}\}$ wykaz outlierów



Etap korelacji

Etap korelacji

Cel etapu: Oszacowanie związków i wzajemnych zależności między wynikami poszczególnych testów algorytmów grafowych, co pozwala zrozumieć współzależności i wpływ różnych czynników na ostateczne rezultaty.



Kroki:

- Wybór metody statystycznej odpowiedniej do analizy korelacji, np. korelacji Pearsona lub Spearmana, zależnie od rozkładu i charakterystyki danych.
- Obliczenie współczynników korelacji dla par zmiennych, co pozwoli na identyfikację silnych i słabych zależności.
- Interpretacja wyników, identyfikacja kluczowych czynników wpływających na efektywność algorytmów i podejmowanie decyzji na podstawie uzyskanych korelacji.

Produkt:

- Wynik korelacji.

Narzędzia i techniki:

- Oprogramowanie statystyczne, jak R czy Python z bibliotekami do analizy danych, które umożliwiają efektywne przeprowadzenie obliczeń.
- Skrypty automatyzujące proces, co pozwala na szybkie ponowne analizy w miarę dostarczania nowych danych testowych.
- Arkusze kalkulacyjne: Microsoft Excel, Google Sheets.

Etap korelacji

Korelacja Pearsona

Kiedy stosować:

- Gdy istnieje potrzeba oceny siły oraz kierunku zależności między zmiennymi.

Zalety:

- Możliwość jasnej interpretacji wyników.
- Efektywność w analizie przyczynowo-skutkowej między zmiennymi.

Wzór:

$$r = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}}$$

gdzie:

X_i, Y_i - są wartościami zmiennych

$\bar{X}, \bar{Y}$ - są średnimi wartościami odpowiednio dla X, Y

Wynik:

$r = 1$ Korelacja dodatnia doskonała.

$r = 0$ Brak liniowej zależności

$r = -1$ Korelacja ujemna doskonała.

Etap korelacji

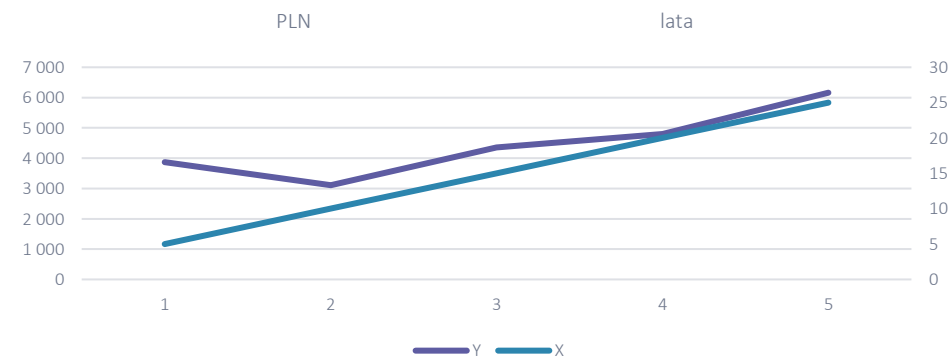
Korelacja Pearsona

Przykład: Analiza związku między latami doświadczenia zawodowego a rocznymi dochodami.

i	Lata doświadczenia X_i	Dochody Y_i	Średnia Lata doświadczenia $\bar{X}$	Średnia Dochody $\bar{Y}$	Odchylenie Lata od Średniej $X_i - \bar{X}$	Odchylenie Dochody od Średniej $Y_i - \bar{Y}$	Iloczyn Odchyłeń $(X_i - \bar{X})(Y_i - \bar{Y})$	Kwadrat Odchylenia Lata $(X_i - \bar{X})^2$	Kwadrat Odchylenia Dochody $(Y_i - \bar{Y})^2$
1	5	3 869	15,00	4 457,40	-10,00	-588,40	5 884,00	100,00	346 214,56
2	10	3 108	15,00	4 457,40	-5,00	-1 349,40	6 747,00	25,00	1 820 880,36
3	15	4 352	15,00	4 457,40	0,00	-105,40	0,00	0,00	11 109,16
4	20	4 800	15,00	4 457,40	5,00	342,60	1 713,00	25,00	117 374,76
5	25	6 158	15,00	4 457,40	10,00	1 700,60	17 006,00	100,00	2 892 040,36

Wynik:

$r=0,87 \sim$ korelacja dodatnia bardzo silna



Etap normalizacji

Etap normalizacji

Cel etapu: dostosowanie wyników testów do wspólnego standardu, umożliwiając ich właściwe porównanie i analizę. Wyskalowanie danych i sformatowanie, aby eliminować wszelkie błędy wynikające z różnic w skalach pomiarowych lub jednostkach.



Kroki:

- Przegląd danych celem zrozumienia ich charakterystyki i identyfikacji ewentualnych anomalii.
- Zależnie od danych i celu analizy, wybierana jest odpowiednia metoda normalizacji, np. normalizacja min-max, standardyzacja Z-score, czy normalizacja według normy.
- Po normalizacji sprawdza się, czy dane zostały poprawnie przetworzone i czy spełniają oczekiwania analizy.

Produkt:

- Znormalizowane wyniki testów.

Narzędzia i techniki:

- Oprogramowanie statystyczne, jak R czy Python z bibliotekami do analizy danych, które umożliwiają efektywne przeprowadzenie obliczeń.
- Skrypty automatyzujące proces, co pozwala na szybkie ponowne analizy w miarę dostarczania nowych danych testowych.
- Arkusze kalkulacyjne: Microsoft Excel, Google Sheets.

Etap normalizacji

Normalizacja min-max

Kiedy stosować:

- Gdy dane mają zróżnicowane zakresy.
- Gdy istnieje potrzeba zdefiniowania danych w stałym zakresie.

Zalety:

- Prostota implementacji.
- Zachowanie kształtu rozkładu danych.

Wynik:

X' – znormalizowany wynik testu.

Wzór:

$$X'_i = \frac{X_i - X_{min}}{X_{max} - X_{min}}$$

gdzie:

X_i - to obserwowana wartość zmiennej,

X_{min} - to minimalna wartość zmiennej w zbiorze danych,

X_{max} - to maksymalna wartość zmiennej w zbiorze danych,

X'_i - to znormalizowana wartość, która będzie w zakresie od 0 do 1.

Etap normalizacji

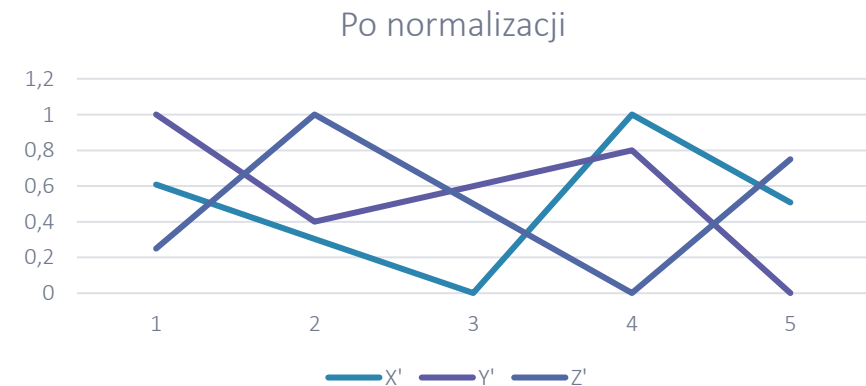
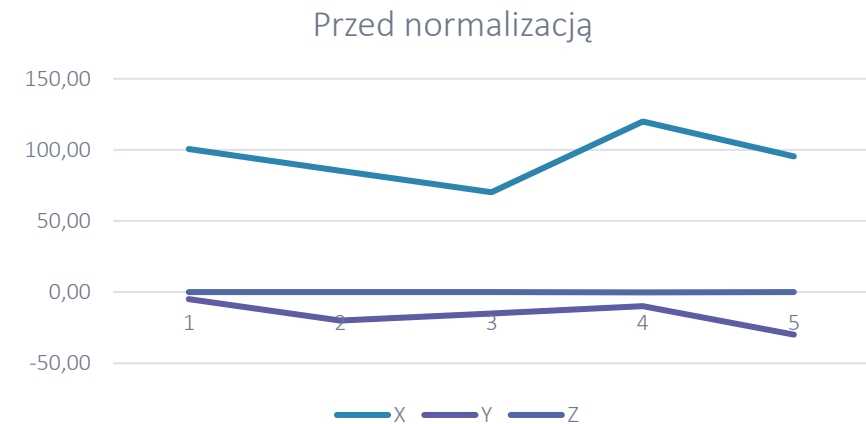
Normalizacja min-max

Przykład: Analiza jakości połączenia sieciowego w różnych lokalizacjach.

Numer lokalizacji i	Przepustowość (mbps) X_i	Czas opóźnienia (ms) Y_i	Utrata pakietów (%) Z_i	X_{min}	X_{max}	Y_{min}	Y_{max}	Z_{min}	Z_{max}	X'	Y'	Z'
1	100,50	-5	0,02	70,2	120	-30	-5	0,01	0,05	0,6084	1	0,25
2	85,30	-20	0,05	70,2	120	-30	-5	0,01	0,05	0,3032	0,4	1
3	70,20	-15	0,03	70,2	120	-30	-5	0,01	0,05	0	0,6	0,5
4	120,00	-10	0,01	70,2	120	-30	-5	0,01	0,05	1	0,8	0
5	95,50	-30	0,04	70,2	120	-30	-5	0,01	0,05	0,508	0	0,75

Wynik:

X', Y', Z' znormalizowane wyniki testów



Etap rankingu

Etap rankingu

Cel etapu: opracowanie uporządkowanej listy zagregowanych wyników na podstawie znormalizowanych danych wynikowych. Ranking umożliwia identyfikację najlepszych i najgorszych analizowanych danych, dostarczając istotnych informacji, które mogą wpłynąć na podejmowanie decyzji oraz strategię dalszego rozwoju i interwencji.



Kroki:

- Określenie, które zmienne lub kombinacje zmiennych będą brane pod uwagę przy ustalaniu rankingu.
- Wykorzystanie danych, które zostały uprzednio znormalizowane do oszacowania wartości z różnymi charakterami (stymulanty, destymulanty, nominanty).
- Agregacja oszacowanych wartości z uwzględnieniem wag specyficznych dla określonych testów.

Produkt:

- Ranking wyników.

Narzędzia i techniki:

- Oprogramowanie statystyczne, jak R czy Python z bibliotekami do analizy danych, które umożliwiają efektywne przeprowadzenie obliczeń.
- Skrypty automatyzujące proces, co pozwala na szybkie ponowne analizy w miarę dostarczania nowych danych testowych.
- Arkusze kalkulacyjne: Microsoft Excel, Google Sheets.

Etap rankingu

Szacowanie rankingu wielokryterialnego z uwzględnieniem stymulant, destymulant i nominant oraz wag

Kiedy stosować:

- Gdy ocena obejmuje różnorodne typy kryteriów.
- Gdy istnieje potrzeba ujednoliconego podejścia do oceny danych o różnym charakterze i znaczeniu (wadze).

Dla stymulant (gdzie wyższe wartości są lepsze):

$$X''_{ij} = X'_{ij}$$

Dla destymulant (gdzie niższe wartości są lepsze):

$$X''_{ij} = 1 - X'_{ij}$$

Dla nominant (gdzie istnieje optymalna wartość):

$$X''_{ij} = 1 - |X'_{ij} - X_{opt,j}|$$

gdzie:

X'_{ij} - to znormalizowana wartość i – *tej* zmiennej j – *tego* testu,

$X_{opt,j}$ - to wartość optymalna (po normalizacji) j – *tego* testu,

w_j - waga dla j – *tego* testu gdzie $w_j > 0$ i $\sum_j w_j = 1$.

Ranking końcowy jest obliczany przez ważenie i sumowanie stymulant, destymulant i nominant:

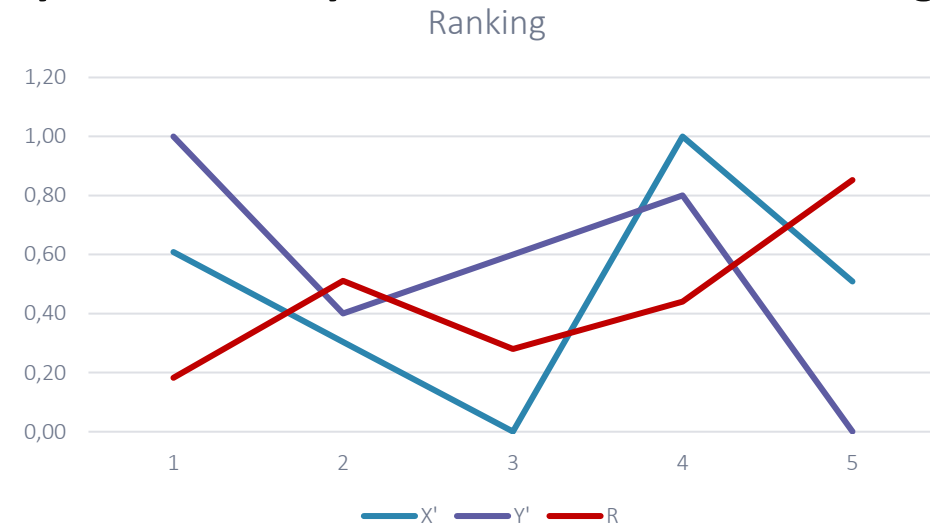
$$R_i = \sum_{j=1}^n w_j \times X''_{ij}$$

Etap rankingu

Szacowanie rankingu wielokryterialnego z uwzględnieniem stymulant, destymulant i nominant oraz wag

Przykład: Ranking lokalizacji na podstawie parametrów sieci.

Numer lokalizacji i	Przepustowość (stymulanta) X'_i	Czas opóźnienia (destymulanta) Y'_i	X''_i	Y''_i	w_x	w_y	Wartość w rankingu R_i	pozycja
1	0,61	1	0,61	0	0,3	0,7	0,18	5
2	0,30	0,4	0,30	0,6	0,3	0,7	0,51	2
3	0,00	0,6	0,00	0,4	0,3	0,7	0,28	4
4	1,00	0,8	1,00	0,2	0,3	0,7	0,44	3
5	0,51	0	0,51	1	0,3	0,7	0,85	1



Wynik:

R ranking zagregowanych wyników

Etap grupowania

Etap grupowania

Cel etapu: podział wyników w rankingu na grupy między innymi określające zwycięzców oraz przegranych.

Kroki:

- Sortowanie poprzez uporządkowanie wszystkich wyników w rankingu od najwyższego do najniższego.
- Podział pozycji na istotne grupy między innymi wskazujące zwycięzców oraz przegranych z wykorzystaniem metod statystycznych takich jak: kwartyle, percentyle, proste progi, odchylenie standardowe lub podział na podstawie kryteriów zdefiniowanych przez ekspertów.
- Analiza uzyskanych podziałów.

Produkt:

- Ranking podzielony na grupy.

Narzędzia i techniki:

- Oprogramowanie statystyczne, jak R czy Python z bibliotekami do analizy danych, które umożliwiają efektywne przeprowadzenie obliczeń.
- Skrypty automatyzujące proces, co pozwala na szybkie ponowne analizy w miarę dostarczania nowych danych testowych.
- Arkusze kalkulacyjne: Microsoft Excel, Google Sheets.



Etap grupowania

Percentyle

dzielą zestaw danych na sto równych części, co pozwala na ocenę relatywnej pozycji poszczególnych wyników.

$$k = \frac{P}{100} \times (n + 1)$$

gdzie:

n – liczba wyników,

P – pożądaný percentyl (od 1 do 100),

k - pozycja w rankingu dla wskazanego percentyla.

Wyniki:

Jeśli wyniki są posortowane od najmniejszej do największej pozycji, gdzie najmniejsza pozycja to najlepszy wynik, a największa pozycja to najgorszy wynik, to:

Zwycięzcy to wyniki niższe lub równe niż wartość dla 10. percentyla ($P = 10$)

Przegrani to wyniki wyższe lub równe niż wartość dla 90. percentyla ($P = 90$)

Etap grupowania

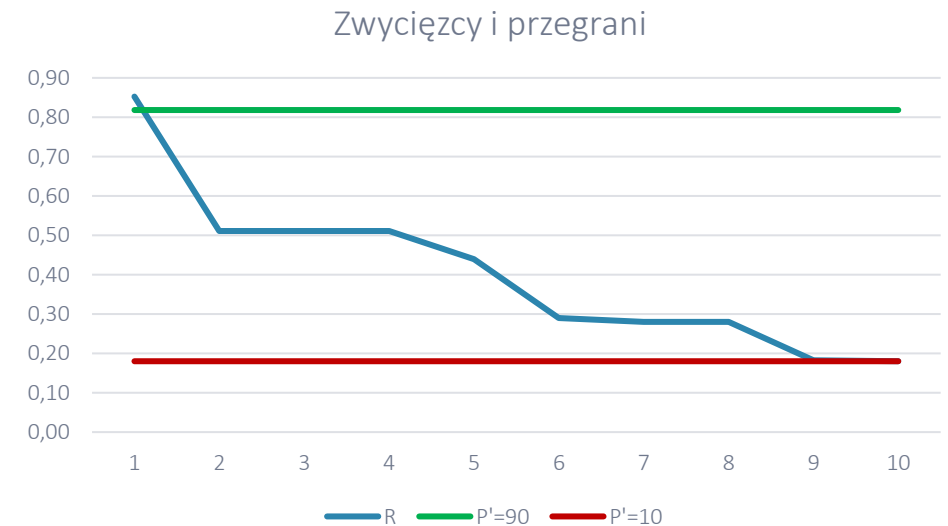
Percentyle

Przykład: Określenie grup najlepszych i najgorszych lokalizacji sieci.

Numer lokalizacji i	Wartość w rankingu R_i
1	0,18
2	0,51
3	0,28
4	0,44
5	0,85
6	0,51
7	0,28
8	0,18
9	0,29
10	0,51

Sortowanie
pozycji

Numer lokalizacji i	Wartość w rankingu R_i	pozycja	
5	0,85	1	Zwycięzca
2	0,51	2	
6	0,51	3	
10	0,51	4	
4	0,44	5	
9	0,29	6	
3	0,28	7	
7	0,28	8	
1	0,18	9	
8	0,18	10	Przegrany



Wynik:

1,10 - wartość dla 10. percentyla ($P = 10$)

9,90 - wartość dla 90. percentyla ($P = 90$)

Zwycięzcy ($\geq 1,10$) {lokalizacja 5}

Przegrani ($\leq 9,90$) {lokalizacja 8}

Etap centralności

Etap centralności

Cel etapu: oszacowanie miar tendencji centralnej dla zrankingowanych danych, co pozwala zidentyfikować główne trendy, dominujące wzorce oraz centralne punkty danych w zbiorze wyników.



Kroki:

- Zależnie od charakterystyki i rozkładu danych, wybiera się odpowiednią miarę centralności, taką jak średnia, mediana, czy dominanta.
- Na podstawie znormalizowanych i zrankingowanych danych oblicza się wybraną miarę centralności, co daje informacje o "środku" badanego zbioru danych.
- Analizuje się uzyskane wyniki, by zrozumieć, jakie informacje niesie ze sobą centralność analizowanych danych.

Produkt:

- Wartości miar centralności rankingu.

Narzędzia i techniki:

- Oprogramowanie statystyczne, jak R czy Python z bibliotekami do analizy danych, które umożliwiają efektywne przeprowadzenie obliczeń.
- Skrypty automatyzujące proces, co pozwala na szybkie ponowne analizy w miarę dostarczania nowych danych testowych.
- Arkusze kalkulacyjne: Microsoft Excel, Google Sheets.

Etap centralności

Miary centralności (średnia, mediana i dominanta)

- Średnia obliczana jest jako suma wszystkich wartości podzielona przez ich liczbę. Dostarcza informacji o przeciętnej wartości w rankingu.

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

- Mediana to wartość środkowa zestawu danych, gdy są one uporządkowane. Dzieli rozkład na dwie równie liczne grupy. Jest odporna na wartości odstające.

Dla zestawu danych z nieparzystą liczbą obserwacji:

$$\hat{X} = X_{(\frac{n+1}{2})}$$

Dla zestawu danych z parzystą liczbą obserwacji:

$$\hat{X} = \frac{X_{(\frac{n}{2})} + X_{(\frac{n}{2}+1)}}{2}$$

gdzie:

n – to liczba wartości w rankingu

X_i – to i – ta wartość w rankingu

Etap centralności

Miary centralności (średnia, mediana i dominanta)

- Dominanta (moda) to najczęściej występująca wartość w zestawie danych. W kontekście danych rankingu, gdzie wartości mogą być bardziej jednorodne, moda pomoże zidentyfikować najczęściej występującą wartość.

Polega na ręcznym zliczeniu wszystkich wartości w zestawie danych i zidentyfikowaniu wartości, która występuje najczęściej.

Założmy, że mamy następujący zestaw danych {2, 4, 4, 4, 5, 5, 7, 9}

2: 1 raz

4: 3 razy

5: 2 razy

7: 1 raz

9: 1 raz

Dominanta to wartość, która występuje najczęściej, czyli 4.

Etap centralności

Miary centralności (średnia, mediana i dominanta)

Przykład: Określenie średnich wartości rankingu „Ranking lokalizacji” na podstawie parametrów sieci.

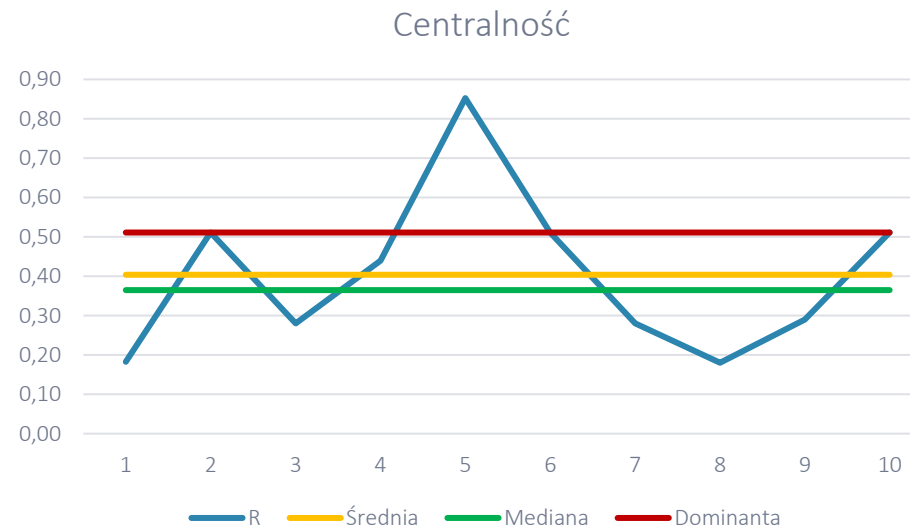
Numer lokalizacji i	Wartość w rankingu R_i
1	0,18
2	0,51
3	0,28
4	0,44
5	0,85
6	0,51
7	0,28
8	0,18
9	0,29
10	0,51

Wynik:

Średnia : 0,40

Mediana : 0,37

Dominanta : 0,51



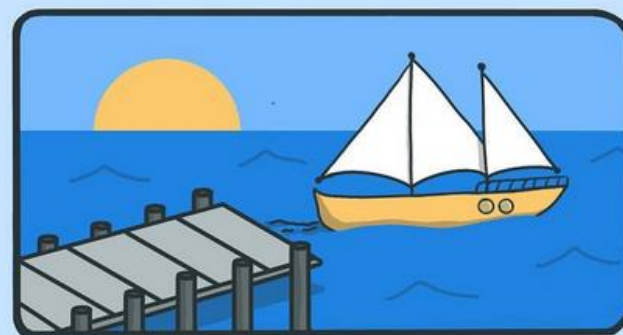
Zadanie: Faza rozszerzonej analizy

Dla przypadku 1. w fazie rozszerzonej analizy:

1. Jakiego wyniku testów jest wskazane aby korelować?
2. Jak obecność outsiderów wpłynie na wyniki?
3. Jakiego wyniku testów będą normalizowane?
4. Którego wyniku testów będą stymulantami, destymulantami i nominantami?
5. Jak dobrać wartości wagi dla znormalizowanych wyników podczas etapu rankingu?
6. Które z miar centralności (średnia, mediana i dominanta) warto wykorzystać?
7. Na jakie grupy warto podzielić wyniki (pozycje) rankingu?



COMFORT ZONE



GROWTH ZONE



Faza prezentacji analizy

ANALIZA DANYCH
Z WYKORZYSTANIEM BAZY
GRAFOWEJ Neo4J
I ALGORYTMÓW GRAFOWYCH

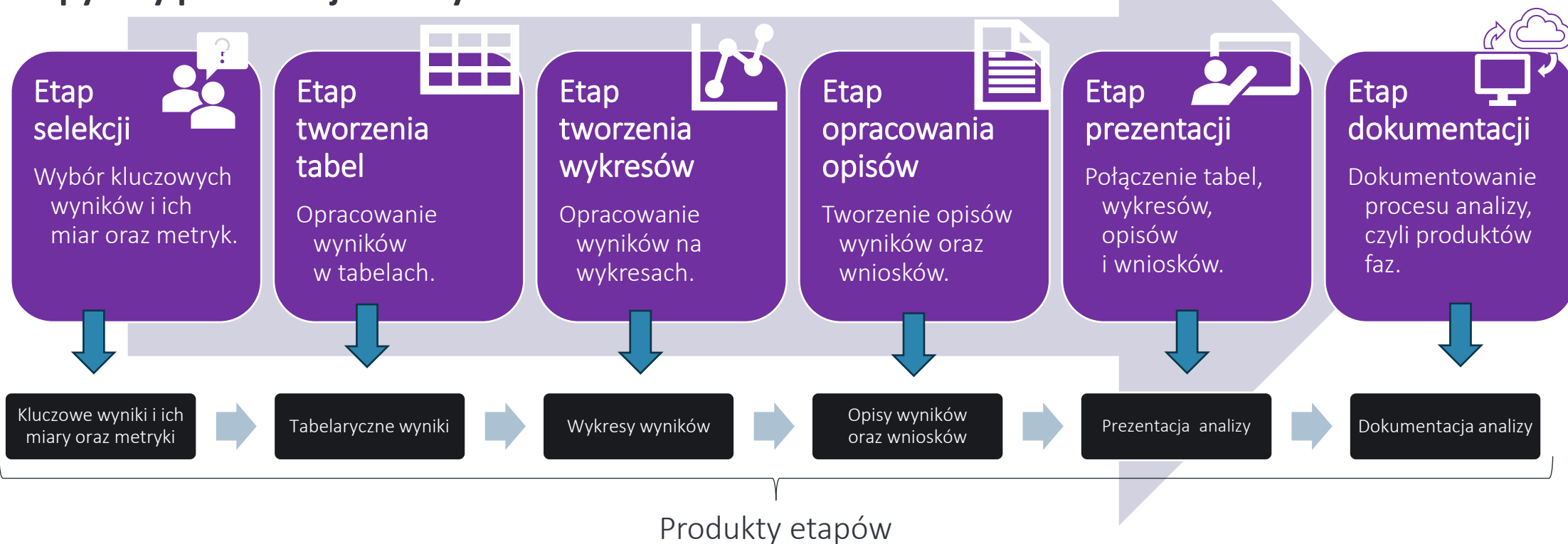


Zagadnienia

- Faza prezentacji analizy
 - Etap selekcji
 - Etap tworzenia tabel
 - Etap tworzenia wykresów
 - Etap opracowania opisów
 - Etap prezentacji
 - Etap dokumentacji
- Zadanie: Faza prezentacji analizy

Faz prezentacji analizy

Etapy fazy prezentacji analizy



Etap selekcji

Etap selekcji

Cel etapu: zidentyfikowanie i wyodrębnienie najbardziej znaczących informacji z kompleksowego zestawu danych. Dzięki temu możliwe jest skoncentrowanie uwagi odbiorców na najbardziej istotnych wynikach i ułatwienie zrozumienia przekazywanych informacji.



Kroki:

- Przegląd całego zbioru wyników w celu zrozumienia zakresu danych i ich znaczenia.
- Ustalenie, które wyniki są kluczowe dla celów analizy, na podstawie wcześniej zdefiniowanych celów.
- Wybór danych, które najlepiej reprezentują osiągnięte wyniki i są najbardziej istotne.
- Decyzja o tym, które mierniki (np. wartości średnie, korelacja, ranking itd.) najlepiej ilustrują wyniki i zostaną użyte do dalszej prezentacji.

Produkt:

- Kluczowe wyniki oraz ich miary i metryki.

Narzędzia i techniki:

- Oprogramowanie statystyczne, jak R czy Python z bibliotekami do analizy danych, które umożliwiają efektywne przeprowadzenie obliczeń.
- Skrypty automatyzujące proces, co pozwala na szybkie ponowne analizy w miarę dostarczania nowych danych testowych.
- Arkusze kalkulacyjne: Microsoft Excel, Google Sheets.

Etap tworzenia tabel

Etap tworzenia tabel

Cel etapu: Przygotowanie tabel do klarownego prezentowania wyników, co ułatwia ich interpretację i porównanie. Tabele umożliwiają szybkie zrozumienie danych poprzez uporządkowane i strukturalne przedstawienie informacji.



Kroki:

- Wybór danych, które zostaną zaprezentowane w tabelach. To obejmuje kluczowe metryki i wyniki, które zostały wcześniej wyselekcjonowane jako istotne.
- Decyzja o strukturze tabel, które najlepiej oddadzą charakter i znaczenie danych (np. tabela przestawna, tabela z agregatami).
- Ustalenie formatowania tabel, w tym nagłówków, kolumn i rodzajów danych, aby ułatwić odbiór informacji.
- Sprawdzenie, czy wszystkie dane są dokładnie i prawidłowo wprowadzone do tabel, weryfikacja poprawności liczb i etykiet.

Produkt:

- Tabelaryczne wyniki.

Narzędzia i techniki:

- Oprogramowanie statystyczne, jak R czy Python z bibliotekami do analizy danych, które umożliwiają efektywne przeprowadzenie obliczeń.
- Skrypty automatyzujące proces, co pozwala na szybkie ponowne analizy w miarę dostarczania nowych danych testowych.
- Arkusze kalkulacyjne: Microsoft Excel, Google Sheets.

Etap tworzenia wykresów

Etap tworzenia wykresów

Cel etapu: Wizualizacja wyników poprzez wykresy w celu lepszego zrozumienia trendów, wzorców i korelacji w zestawie danych. Wykresy ułatwiają szybką interpretację danych, co jest kluczowe w prezentacjach.



Kroki:

- Decyzja o rodzajach wykresów, które najlepiej oddają charakter wyników i cel analizy (np. wykres słupkowy, liniowy, kołowy).
- Wykorzystanie tabelarycznych wyników do budowy wykresów.
- Użycie narzędzi do stworzenia wykresów, włączając w to ustawienia estetyczne takie jak kolory, etykiety, legendy i tytuły.
- Dostosowanie skal (liniowa, procentowa, logarytmiczna, kwantylowa itd..), zakresów i innych elementów wykresów w celu poprawy czytelności, porównywania i zrozumienia prezentowanych danych.
- Sprawdzenie, czy wykresy są zrozumiałe i czy poprawnie prezentują wyniki.

Produkt:

- Wykresy wyników.

Narzędzia i techniki:

- Oprogramowanie statystyczne, jak R czy Python z bibliotekami do analizy danych, które umożliwiają efektywne przeprowadzenie obliczeń.
- Skrypty automatyzujące proces, co pozwala na szybkie ponowne analizy w miarę dostarczania nowych danych testowych.
- Arkusze kalkulacyjne: Microsoft Excel, Google Sheets.

Etap opracowania opisów

Etap opracowania opisów

Cel etapu: opracowanie precyzyjnych opisów wyników i wniosków z analizy, które klarownie komunikują kluczowe informacje i interpretacje w powiązaniu z celami.



Kroki:

- Przeprowadzenie głębokiej analizy wyników w celu pełnego zrozumienia uzyskanych danych i ich znaczenia.
- Identyfikacja głównych wniosków, które można wyciągnąć z analizy, i które są istotne dla osiągnięcia celów.
- Stworzenie opisów, które precyzyjnie odnoszą się do danych, wyników i ich analizy, zapewniając jasność i dokładność w komunikowaniu.
- Przejrzenie napisanych opisów pod kątem poprawności językowej, spójności logicznej oraz adekwatności do prezentowanych danych.
- Upewnienie się, że opisy są odpowiednio zintegrowane z tabelami, wykresami i innymi elementami wizualnymi, aby wzajemnie na siebie oddziaływały i się uzupełniały.

Produkt:

- Opisy wyników i wnioski.

Narzędzia i techniki:

- Edytory tekstowe takie jak: Microsoft Word lub Google Docs do pisania i edytowania tekstów.
- Narzędzia do korekty tekstu jak: LanguageTool lub Grammarly, aby zapewnić poprawność gramatyczną i stylistyczną opisów.
- Wykorzystanie narzędzi takich jak: Microsoft Teams, Slack czy Google Workspace, które umożliwiają współpracę i łatwe udostępnianie dokumentów między członkami zespołu.

Etap prezentacji

Etap prezentacji

Cel etapu: integracja wszystkich elementów analizy — danych, wyników, tabel, wykresów, opisów i wniosków — w jednolitą, spójną prezentację, która efektywnie komunikuje główne odkrycia i poziom osiągnięcia celów analizy.



Kroki:

- Zbieranie wszystkich wcześniej przygotowanych komponentów analizy, takich jak tabele, wykresy i tekstowe opisy wyników.
- Ułożenie materiałów w logicznej kolejności, która opowiada historię danych od wprowadzenia do wniosków końcowych i określenia poziomu osiągnięcia celów analizy.
- Projektowanie układu prezentacji, w tym decyzje dotyczące rozmieszczenia treści na stronach dokumentu lub slajdach prezentacji, aby maksymalnie zwiększyć czytelność i przyciągnąć uwagę odbiorców.
- Upewnienie się, że wszystkie elementy wizualne są odpowiednio zintegrowane z opisami tekstowymi, co obejmuje dopasowanie legend wykresów do tekstu, wyjaśnienia tabel i odniesienia do nich w opisach.
- Sprawdzanie całości prezentacji pod kątem spójności, poprawności merytorycznej i stylistycznej oraz dokonywanie niezbędnych korekt.

Produkt:

- Prezentacja analizy.

Narzędzia i techniki:

- Narzędzia do Projektowania Graficznego: Programy takie jak Adobe InDesign lub Microsoft Publisher, które pozwalają na profesjonalne projektowanie layoutów prezentacji.
- Oprogramowanie Prezentacyjne: Narzędzia takie jak Microsoft PowerPoint, Google Slides czy Prezi, które są używane do tworzenia dynamicznych oraz interaktywnych prezentacji.
- Narzędzia do Integracji Danych: Platformy takie jak Tableau lub Power BI, które mogą być używane do tworzenia interaktywnych dashboardów łączących wykresy, tabele i tekst w jednym miejscu.

Etap dokumentacji

Etap dokumentacji

Cel etapu: dokumentowanie całego procesu analizy oraz produktów poszczególnych etapów, w celu zapewnienia pełnej przejrzystości, umożliwienia weryfikacji oraz ułatwienia odniesienia do analizy w przyszłości.



Kroki:

- Zbieranie produktów z poprzednich etapów. Kompilacja wszystkich dokumentów, danych surowych, wykresów, tabel i opisów, które zostały wygenerowane w poszczególnych etapach.
- Szczegółowe opisanie użytych metod, procedur analizy danych, wykorzystanych narzędzi oraz kryteriów decyzyjnych, co umożliwia zrozumienie procesu i powtarzalność badania.
- Opracowanie dokumentu podsumowującego główne wyniki analizy, wnioski, poziom osiągnięcia celów, a także potencjalne zalecenia i kierunki dalszych działań.
- Zorganizowanie dokumentacji w sposób uporządkowany, co obejmuje cyfrową archiwizację i fizyczne przechowywanie dokumentów.

Produkt:

- Dokumentacja analizy.

Narzędzia i techniki:

- Systemy Zarządzania Dokumentacją: Narzędzia takie jak Microsoft SharePoint lub systemy ECM (Enterprise Content Management), które pozwalają na bezpieczne przechowywanie, łatwy dostęp i zarządzanie dokumentacją.
- Narzędzia do Tworzenia Dokumentów: Programy takie jak Microsoft Word lub Google Docs, używane do tworzenia i formatowania dokumentacji tekstowej.
- Oprogramowanie do Zarządzania Projektami: Narzędzia takie jak Jira, Asana czy Trello, które mogą być wykorzystywane do śledzenia postępów i dokumentowania różnych etapów projektu.

Zadanie: Faza prezentacji analizy

Dla przypadku 1. w fazie prezentacji analizy:

1. Jakie elementy należy włączyć do prezentacji końcowej, aby była ona kompletna i spójna?
2. Jakie strategie można zastosować, aby efektywnie połączyć tabele, wykresy i opisy w jednolity dokument lub prezentację?
3. Jak dostosować prezentację do różnych grup odbiorców, np. zarząd firmy, potencjalnych klientów lub partnerów naukowych?
4. Jakie są najlepsze praktyki w zakresie prezentowania złożonych danych analitycznych, aby były one zrozumiałe dla osób niebędących ekspertami w danej dziedzinie?
5. W jaki sposób można wykorzystać interaktywne elementy, takie jak dashboardy czy interaktywne wykresy, w prezentacji wyników analizy?
6. Jak zaprezentować poziom osiągnięcia celów analizy?
7. Jak zachować odpowiedni balans w prezentacji pomiędzy pokazaniem działań podejmowanych podczas analizy, wynikami, wnioskami i osiągniętymi celami?

INFORMATION
20 YEARS AGO



INFORMATION
TODAY



ROBERTOFERRARO.ART

