Sprawozdanie z projektu zaliczeniowego z Sieci Złożonych

Temat: Sieć restauracji we Wrocławiu

Adam Bednarski 260338

# 1. Wstęp

## 1.1 Temat

Tematem mojego projektu jest wizualizacja i analiza sieci wrocławskich restauracji, w szczególności tych, które swoje usługi „dowozowe” świadczą za pośrednictwem jednej z najpopularniejszych w Polsce firm oferujących taką usługę, czyli Uber Eats. Z zasugerowanych przez prowadzącego kurs dwóch „ścieżek działania” wybrałem ścieżkę symulacyjną. Rozszerza to więc mój temat o kolejne zagadnienie, czyli zasymulowanie funkcjonowania restauracji we Wrocławiu w celu sprawdzenia, które aktualnie działające restauracje mają szansę utrzymać się na rynku restauracji w ciagu najbliższych 5 lat. Szczegóły wizualizacji, analizy i symulacji opiszę w osobnych podpunktach. Ponadto, z uwagi na ogromną liczbę restauracji na terenie Wrocławia, restauracje zawarte w mojej sieci zostały ograniczone przeze mnie do tych, które oferują najpopularniejsze według mnie typy jedzenia.

## 1.2 Środowisko i organizacja pracy

Projekt realizowałem przez drugą połowę semestru, progresywnie go rozwijając i rozbudowując. Bieżące problemy rozwiązywałem samodzielnie po uprzedniej konsultacji tych problemów z prowadzącym. Do implementacji i wizualizacji mojej sieci wykorzystałem język programowania Python, głównie przy użyciu bibliotek:

a) pyvis, matplotlib oraz networkx do wizualizacji sieci

b) BeautifulSoup do ekstrakcji danych ze strony internetowej UberEats

c) numpy oraz networkx do analizy sieci

## 2. Cel projektu

Wybierając ścieżkę symulacyjną zdecydowałem się sprawdzić, które z obecnie funkcjonujących we Wrocławiu restauracji w przyszłości mają szanse na największy sukces. Oczywiście spośród wielu czynników mogących mieć na to wpływ ja wybrałem ten, który najbardziej według mnie oddaje prestiż i pozycję restauracji na rynku, czyli jej ranking (w tym przypadku ocena na stronie UberEats wystawiana przez klientów). Ponadto, celem jest też analiza wybranych wielkości opisujących sieć złożoną, również tylko tych, które według mnie są najbardziej istotne. Po trzecie, celem jest wizualizacja sieci, dla lepszego zobrazowania rzeczywistości i zmian zachodzących w czasie symulacji.

# 3. Struktura sieci

Sieć złożona powinna spełniać trzy główne założenia:

1. powinna mieć węzły (odpowiedniki wierzchołków w grafie)

2. powinna mieć połączenia (odpowiedniki krawędzi w grafie)

3. powinna być dynamiczna (zdolność do zmian, ewolucji)

Kierując się tymi założeniami zaimplementowałem następującą sieć:

1. Węzłami sieci są restauracje działające na terenie Wrocławia (w szczególności te, które zdecydowałem się ekstraktować ze strony UberEats, zawężając kryterium ekstrakcji do kilku najważniejszych typów dań serwowanych przez restauracje. Każdy węzeł jest unikalny i reprezentowany przez nazwę restauracji, przykładowo „Pizza Hut”.

2. Węzły sieci są połączone na podstawie tego samego kryterium, którego użyłem do ekstrakcji danych, czyli na podstawie typów serwowanych dań. Przykładowo, jeśli restauracja A serwuje danie „kebab” i restauracja B serwuje danie „kebab” to węzeł A będzie połączony z węzłem B. W szczególności, jeśli restauracja A serwuje dania „kebab, pizza”, a restauracja B serwuje dania „makaron, burger, pizza”, to takie węzły będą również połączone.

3. Elementem dynamiki mojej sieci jest ocena danej restauracji, w przypadku UberEats wynosi ona nie więcej niż 5.0. Przykładowo, restauracja Pizza Hut może mieć początkową ocenę 4.6, a pod koniec symulacji ocenę 5.0.

Podsumowując, moją sieć mogę utożsamić z grafem prostym nieskierowanym, dla którego każdy wierzchołek reprezentowany jest przez tytuł restauracji, każda krawędź reprezentowana jest przez połączenia między restauracjami, a atrybutami wierzchołków są typy serwowanych dań i ocena restauracji, z czego ocena jest w mojej sieci elementem dynamicznym.

# 4. Implementacja

Dane ekstrahowałem w następujący sposób:

1. ze strony <https://www.ubereats.com/pl/category/wroclaw-dolnoslaskie> odczytałem i wybrałem następujące kategorie:

*'grill', 'noodles', 'pizza', 'sandwich', 'pollo', 'candy'.*

2. struktura adresu *https* strony UberEats pozwala wyświetlać wszystkie restauracje oferujące daną kategorię jedzenie poprzez dodanie na końcu adresu nazwy kategorii, przykładowo „<https://www.ubereats.com/pl/category/wroclaw-dolnoslaskie/grill>”. Dla każdej z kategorii pobrałem po 30 restauracji serwujące daną kategorię jedzenia. Jeśli restauracja serwowała więcej niż jedną z wybranych przeze mnie kategorii, to nie została dodawana ponownie, zapisywana została jedynie informacja o kolejnym typie kategorii, która serwuje ta restauracja. Dla każdej z restauracji zapisywałem: nazwę restauracji, kategorie serwowanych przez nią dań oraz ocenę restauracji (w przypadku nowo powstałych restauracji na stronie UberEats zamiast oceny liczbowej widnieje napis „Nowość”, dla uproszczenia działania symulacji nadawałem takim restauracjom ocenę 4.5 jako domyślną).

## 4.1 Symulacja

Przechodząc do szczegółów samej implementacji, zacznę od symulacji. Dla wygenerowanej listy restauracji oraz macierzy połączeń między nimi przeprowadzam symulację jak następuje:

2. Parametry stałe (określone deterministycznie):

* sim\_time = 36 – liczba iteracji (1 iteracja odpowiada jednemu mięsiącowi)
* rating[i] – ocena i-tej restauracji [2.0, 5.0], 0 < i < 100
* cuisines[i] – lista typów dań serwowanych przez i-tą restaurację,   
  0 < i < 100

3. Parametry losowe:

* rat\_down[i][j] – wartość, o którą obniży się ocena i-tej restauracji w j-tej iteracji [0,1], 0 < i < 100, 0 < j < 25
* rat\_up[k][j] – wartość, o którą podwyższy się ocena k-tej restauracji w j-tej iteracji [0,1], 0 < k < 100, 0 < j < 25

4. Zasada zmian ocen restauracji i-tej i k-tej w j-tej iteracji:

W każdej j-tej iteracji symulacji porównywana jest ocena każdej i-tej restauracji z każdą oceną jej k-sąsiadniej restauracji (restauracje są sąsiednie jeśli serwują przynajmniej jeden ten sam typ dań). Następnie dla restauracji mającej wyższą ocenę dodawana jest wartość rat\_up, a od niższej – odejmowana wartość rat\_down. Jeśli ocena którejś z restauracji spadnie poniżej 2.0, wówczas restauracja taka zostaje usuwana, a wraz z nią wszystkie jej połączenia z innymi restauracjami. Można więc powiedzieć, że symulacja w pewnym sensie opiera się na zjawisku starzenia się węzłów, a konkretnie jego wariacji, gdzie również co iterację może zostać usunięty jeden (lub zero lub więcej węzłów).

## 4.2 Wizualizacja

Zaimplementowaną przeze mnie sieć wizualizuję na dwa sposoby. Pierwszy sposób to użycie biblioteki pyvis, która pozwala na dużo estetyczniejsze i funkcjonalne wyświetlenie sieci. Powodem takiej wizualizacji jest pomoc w zobrazowaniu sobie aktualnej sytuacji rynkowej restauracji we Wrocławiu, z wyszczególnieniem ocen tych restauracji oraz typów serwowanych przez nie dań. Jednakże z uwagi na wymagającą obliczeniowo i czasowo wizualizację, zdecydowałem się na drugi sposób. Stosuję go by zobrazować zmiany zachodzące w symulacji, iteracja po iteracji. Sposób ten jest uboższy, ponieważ biblioteka networkx nie oferuje tylu funkcji.

## 4.3 Analiza sieci

Zdecydowałem się zbadać trzy wielkości służące do opisania sieci:

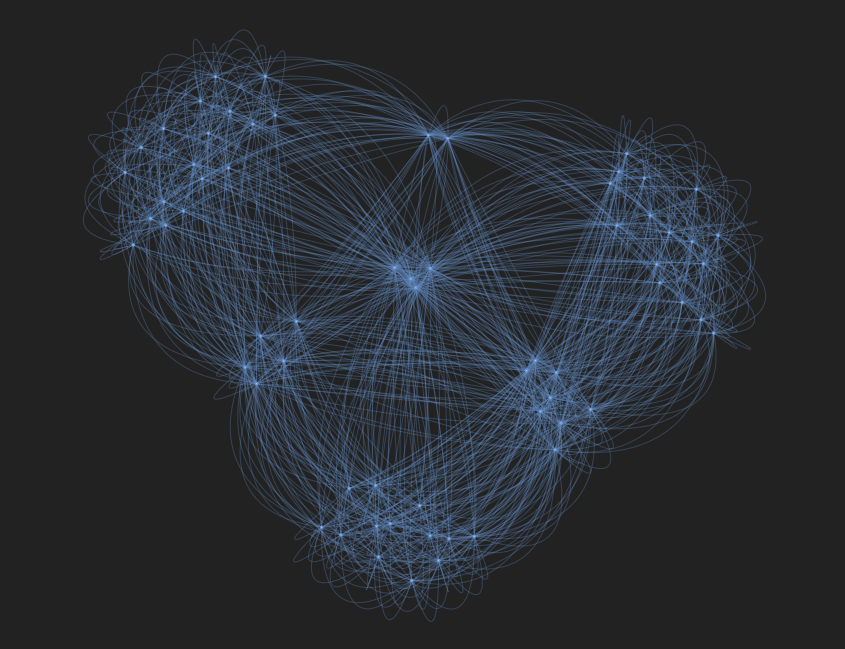
1. Bliskość (ang. closeness centrality) – jedna z miar centralności wierzchołków, równająca się odwrotności średniej odległości wierzchołka od innych wierzchołków. W przypadku mojego projektu jest to lista wartości średnich odległości każdej restauracji od wszystkich innych restauracji. W bardziej potocznym języku miara ta pozwala określić łatwość dostania się z konkretnego węzła do innych węzłów oraz jak dobrym kandydatem do centrum dystrybucji jest dany wierzchołek. Ponownie, w przypadku mojego projektu miara ta może świadczyć o tym, jak duża jest szansa, że klient zdecyduje się na inną alternatywę (restaurację) wybierając inną. Może to też świadczyć o aspekcie marketingowym, jaka jest szansa że klient danej restauracji będzie w stanie znaleźć inną restaurację o potencjalnie podobnym menu.

2. Stopień wierzchołków (ang. node degree) – kolejna z miar centralności wierzchołków, równająca się liczbie połączonych z nim bezpośrednio innych wierzchołków. W przypadku mojego projektu jest to liczba restauracji bezpośrednio połączonych z każdą inną (restauracje serwujące przynajmniej jeden ten sam typ dań). Miara ta pozwoli określić, jak mocno skupiona jest sieć czyli w przełożeniu na rzeczywistość – ile restauracji jest ze sobą w bezpośredniej, bliskiej konkurencji.

3. Średnica grafu (ang. graph diameter) – wielkość charakteryzująca graf oznaczająca długość najdłuższej z najkrótszych ścieżek. Innymi słowy pokazuje, ile wynosi najgorsze skomunikowanie w grafie pomiędzy dwoma różnymi wierzchołkami. W przypadku mojego projektu średnica oznaczała będzie dwie najdalej oddalone od siebie restauracje (w kontekście serwowanych dań). W pewnym sensie średnica będzie oznaczała jak szeroko rozwinięta jest sieć restauracji we Wrocławiu.

# 5. Wyniki

Program zwraca pełne wyniki co 6 iteracji (6 miesięcy czasu rzeczywistego), ale na potrzeby dalszych wniosków przeanalizuję wyniki przed i po wykonaniem symulacji:



Rys 1. Wizualizacja sieci przed symulacją

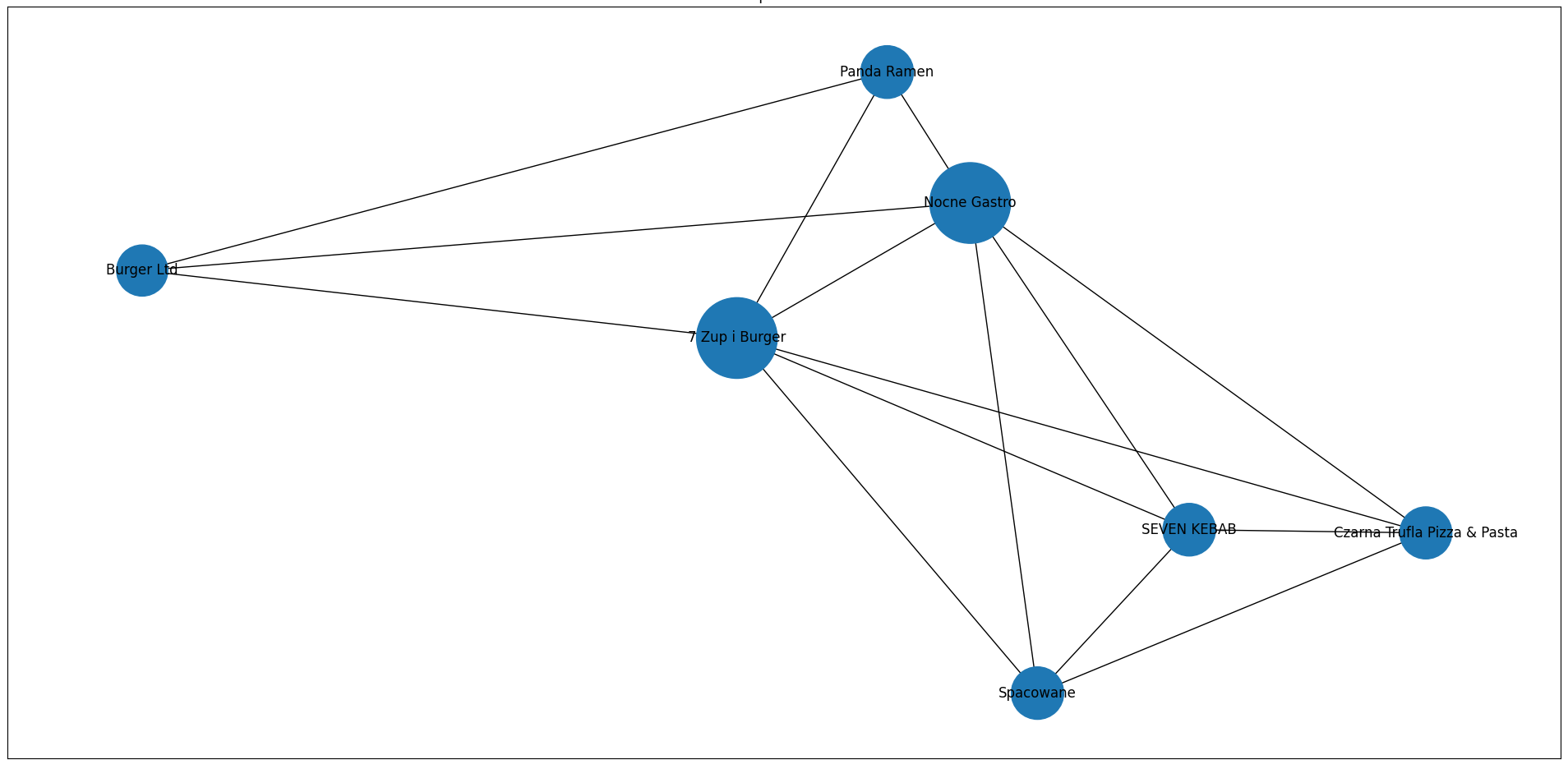
Patrząc na wizualizację grafu przed symulację bardzo jasno widać zjawisko małych światów, czyli bardzo blisko połączonych ze sobą skupisk węzłów. Jest to zjawisko rzeczywiście występujące w rzeczywistości, gdzie restauracje „dzielą się” na te serwujące te same typy dań. Co więcej, duża część węzłów posiada ten sam stopień, co świadczy o ich tendecji do budowania połączeń z podobnie usieciowionymi węzłami. Jest to wyznacznik wysokiej korelacji tych węzłów.

**Średnica grafu: 2**

**Średnia bliskość węzłów: 0.6981**

**Minimalny stopień wierzchołka: 28**

**Maksymalny stopień wierchołka: 65**

****

Rys 3. Wizualizacja sieci po 60 iteracjach

**Średnica grafu: 2**

**Średnia bliskość węzłów: 0.7976**

**Minimalny stopień wierzchołka: 3**

**Maksymalny stopień wierzchołka: 6**

# 6. Analiza i wnioski

Celem analizy sieci restauracji we Wrocławiu było wyłonienie tych restauracji, które potencjalnie za kilka lat utrzymają się na rynku i będą dalej prosperowały. Celami pobocznymi była analiza sieci złożonej pod kątem wielkości jej charakteryzujących, czyli zbadanie średnicy grafu, a także bliskości i stopni wierzchołków.

Badanie zakończyło się sukcesem, zarówno symulacja najbliższych 5 lat sieci restauracji, jak i analiza samej sieci powiodły się.

**Symulacja:**

a) po zakończeniu symulacji pozostało 7 restauracji:

* 7 Zup i Burger
* Nocne Gastro
* SEVEN KEBAB
* Czarna Trufla Pizza & Pasta
* Spacowane
* Panda Ramen
* Burger Ltd

Jest to w przybliżeniu 10% początkowej liczby restauracji wynoszącej 66. Jest to dość mała liczba, choć spodziewana z uwagi na fakt że w mojej symulacji nie uwzględniałem otwierania się nowych restauracji. Warto zadać sobie pytanie, jakie typy dań serwują poszczególne restauracje, i tak kolejno:

* 7 Zup i Burger: „sandwich”, „pollo” (burger, kurczak)
* Nocne Gastro: „sandwich”, „candy”, „pollo” (burger, desery, kurczak)
* SEVEN KEBAB: „burger”
* Czarna Trufla Pizza & Pasta: „burger”
* Spacowane: “burger”
* Panda Ramen: “burger”, “candy” (burger, desery)
* Burger Ltd: “kurczak”

Okazuje się więc, że zdecydowanie dominującym typem dań są burgery (może zaskakiwać fakt, że niektóre restauracje powinny chociażby po zasguerowaniu się ich nazwą serwować inne typy dań, jednak takie dane widnieją na stronie UberEats i nie zawsze muszą w 100% pokrywać się z rzeczywistością. Sam fakt, że burgery są dominującą kuchnią nie jest zaskakujący, gdyż od niedawna rzeczywiście burgerownie należą do jednych z najchętniej odwiedzanych restauracji. Nie przeprowadzam analizy ocen końcowych po symulacji, gdyż po samych typach dań możemy wnioskować, że wszystkie one będą dla siebie konkurencją, niezależnie od ocen.

**Wielkości charakteryzujące sieć:**

Mając informacje o 7 restauracjach pozostałych pod koniec symulacji, możemy porównać dla nich wartości miar charakteryzujących sieć złożoną:

|  |  |  |
| --- | --- | --- |
| **Bliskość węzłów:** | | |
|  | **Przed** | **Po** |
| 7 Zup i Burger | 0.95238 | 1.0 |
| Nocne Gastro | 1.0 | 1.0 |
| SEVEN KEBAB | 0.68965 | 0.75 |
| Czarna Trufla Pizza & Pasta | 0.64356 | 0.75 |
| Spacowane | 0.64356 | 0.75 |
| Panda Ramen | 0.83333 | 0.66666 |
| Burger Ltd | 0.63725 | 0.66666 |

|  |  |  |
| --- | --- | --- |
| **Stopnie węzłów:** | | |
|  | **Przed** | **Po** |
| 7 Zup i Burger | 49 | 6 |
| Nocne Gastro | 65 | 6 |
| SEVEN KEBAB | 29 | 4 |
| Czarna Trufla Pizza & Pasta | 29 | 4 |
| Spacowane | 29 | 4 |
| Panda Ramen | 52 | 3 |
| Burger Ltd | 28 | 3 |

Średnica grafu zarówno przed, jak i po symulacji wynosiła **2**.

Jeśli chodzi o bliskość węzłów, aż 3 z 7 pozostałych restauracji miały poczatkowo bardzo wysokie wartości bliskości. Wartości pozostałych czterech również nie wynosiły mniej niż 0.5. Może to świadczyć o tym, że im bardziej restauracja jest powiązana z innymi, tym większe ma szanse na przetrwanie na rynku. Może być to kwestia na przykład odpowiedniego dostosowywania się do aktualnych potrzeb rynkowych (co klient najczęściej chciałby zjeść, wychodząc na miasto).

Następnie przechodząc do średnicy grafu, niestety nie można wyciągnąć z niej żadnych sensownych wniosków. Świadczy ona jedynie o tym, że sieć restauracji we Wrocławiu jest bardzo „ściśnięta”. Większość restauracji pozostaje ze sobą w bezpośredniej konkurencji, oferując bardzo podobne typy dań.

Stopnie węzłów nie świadczą również o znaczeniu resturacji na rynku i jej potencjalnej zdolności do przetrwania. Wynika to bezpośrednim związkiem ze średnicą grafu, która wynosi 2. Oznacza to, że jeśli restauracja nie jest połączona z inną bezpośrednio, to maksymalnie dzieli ją tylko jedna, inna restauracja. W takim wypadku możemy mówić o mało pod tym względem złożonej sieci.

Podsumowując, pomimo poprawnie przeprowadzonej symulacji oraz wyliczonych wielkości charaketryzujących sieć ciężko jest jednoznacznie stwierdzić, czy miałe one duży wpływ na przebieg symulacji. Jeśli chodzi o samą sieć restauracji we Wrocławiu to jest ona bardzo gęsto połączona, restauracje pozostają między sobą w ścisłej konkurencji i powinny uważnie dbać o oceny wystawiane przez swoich klientów.