
core • logic


AGH

Narzędzia geoinformatyki w optymalizacji usługi Car Sharing

dr inż. Paweł Skrzyński – Katedra Informatyki Stosowanej AGH, Core Logic Sp. z o.o.
dr inż. Piotr Szwed – Katedra Informatyki Stosowanej, AGH

Kraków, 20.03.2024

Plan prezentacji

Część I – platforma Core Logic

1. Prezentacja platformy
2. Moduł sztucznej inteligencji wspierający optymalizację działania
   - Dane
   - Opis rozwiązania
   - Wdrożenie

Część II – kontynuacja prac badawczych

1. Podział na strefy
2. Optymalizacja
   - model stochastyczny MIPS
3. Estymacja prawdopodobieństwa w sieci drogowej
4. Algorytmy propagacji informacji w grafie sieci drogowej

Część I

O Core Logic

- Ponad **12 lat** na rynku software house
- **50+ pracowników**
- Ponad **7 lat doświadczenia** w e-mobility, carsharing:
  najważniejsze projekty car sharing/vehicle monitoring: Traficar (pierwsza platforma car-sharing w Polsce), innogy GO!, Car Sharing Business, VEHIS, e-mobility: Kathrein GmbH
- **Bon innowacyjny wspólnie z AGH** (zastosowanie metod sztucznej inteligencji w car sharing)

Pierwsza platforma car-sharingowa w Polsce

- **Premiera** w 2016
- **Model: free floating model** (wynajem I zakończenie wynajmu w dowolnym miejscu)
- Dedykowana platforma Core Logic
- **Natywne aplikacje mobilne:** Android, iOS
- **Panel administracyjny/CMS -> web**
- **Ciągły rozwój i utrzymanie**
- **Moduł AI na bazie badań prowadzonych wspólnie z AGH**

Traficar

AGH

EUROPEAN UNION
European Regional Development Fund

core • logic

Pierwszy w pełni elektryczny carsharing w Polsce

- **Wdrożenie w** 2019 (tylko EV, BMW i2)
- **Free floating model**
- Platforma wraz z aplikacjami mobilnymi dostarczona **Core Logic**
- **Webowy panel administracyjny wraz z CMS**
- **Automatyzacja/ułatwienia ładowania EV**

innogy go!

AGH

EUROPEAN UNION
European Regional Development Fund

core • logic

Core Logic w liczbach dzisiaj

- **~12 000** samochodów
- **~6000** wynajmów
- **>1 mln** pobrań aplikacji
- **~500** nowych użytkowników
- **> 100 000 km** dzienny przebieg
- **20 000** przejazdów

core•logic

Po co nam AI?

Ograniczona dostępność pojazdów

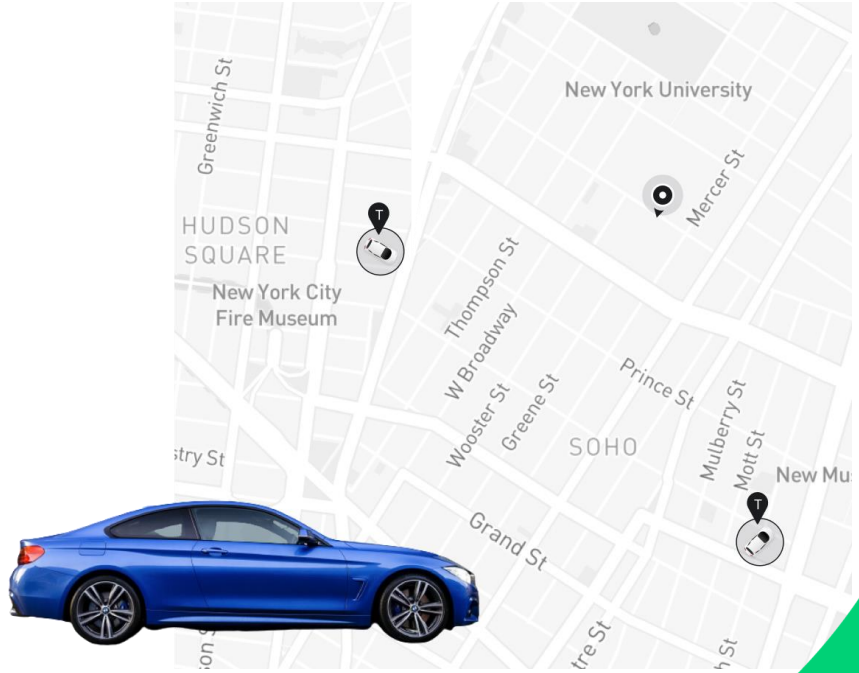
- Dostępność pojazdów – z punktu widzenia użytkownika najważniejszy wskaźnik
- Zależna od lokalizacji pojazdów, ale również platformy software i infrastruktury IT
- Brak pojazdów w strefach wysokiego zapotrzebowania powoduje niedostępność usługi
- Problem z dostarczeniem pojazdów on-demand tam gdzie jest zapotrzebowanie

core • logic

Moduł relokacji

Moduł niezależny możliwy do integracji
z każdym systemem tej klasy.

Moduł przewiduje zapotrzebowanie
i dostępność samochodów
w poszczególnych strefach.



core • logic

Czego potrzebujemy od platformy, żeby zbudować moduł AI?

Założenia modułu AI

- Dane historyczne interakcji użytkowników

- Dane lokalizacyjne pojazdów (historyczne)

- Dane rezerwacji i przejazdów

core•logic

Jak zbudować moduł AI?

Podaż

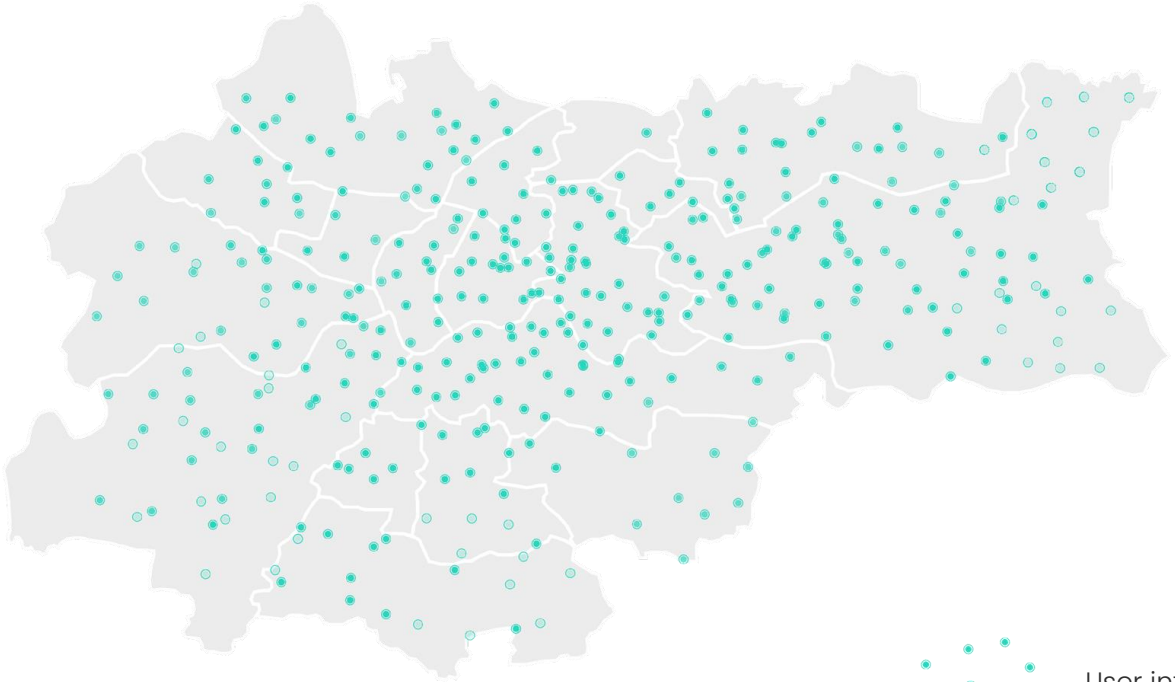
Cars location
in Cracow

AGH

EUROPEAN UNION
European Regional Development Fund

core•logic

Popyt



User interaction
in Cracow

AGH

EUROPEAN UNION
European Regional Development Fund

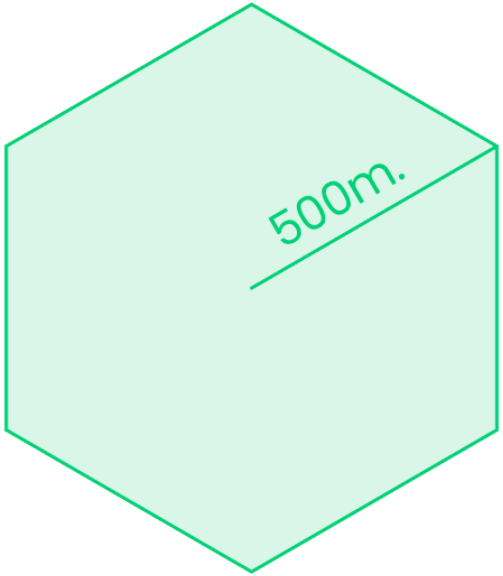
core•logic

Podział stref

Hexagon

Plaster miodu – obszary zbliżone do koła jednak nie nachodzą na siebie i szczelnie wypełniają powierzchnię.

Przekątna 1 km (ok. 10min marszu)



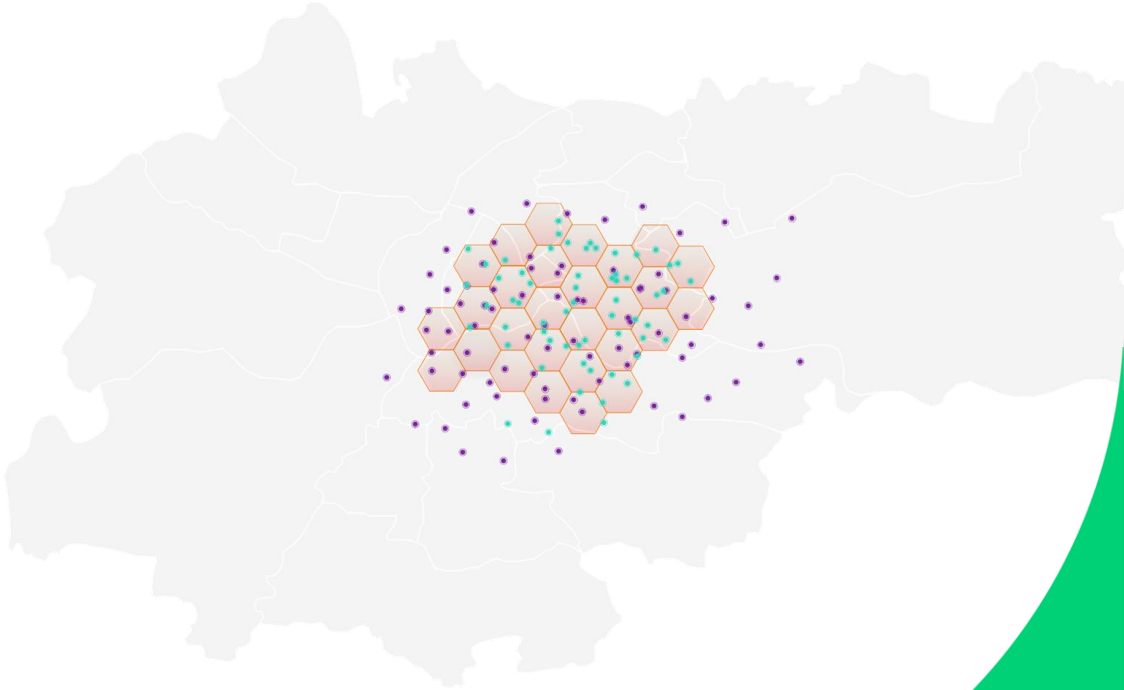
AGH
EUROPEAN UNION
European Regional Development Fund
core•logic

Podaż/popyt analiza

Hexagon

Mapa:

- Interakcje użytkowników – zielone punkty
- Samochody – bordowe punkty



AGH

EUROPEAN UNION
European Regional Development Fund

core•logic

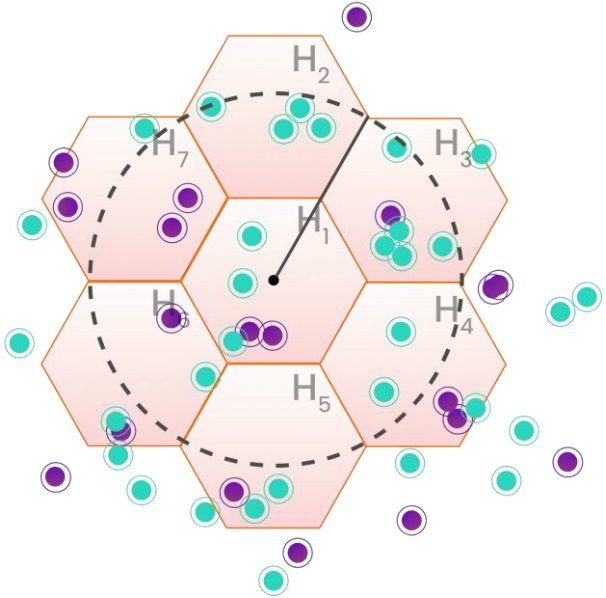
Metodyka - idea

Analiza popytu

Dla każdego heksagonu wyliczany

jest wskaźnik pokrycia popytu w promieniu 1 km

(również sąsiednie heksagony).



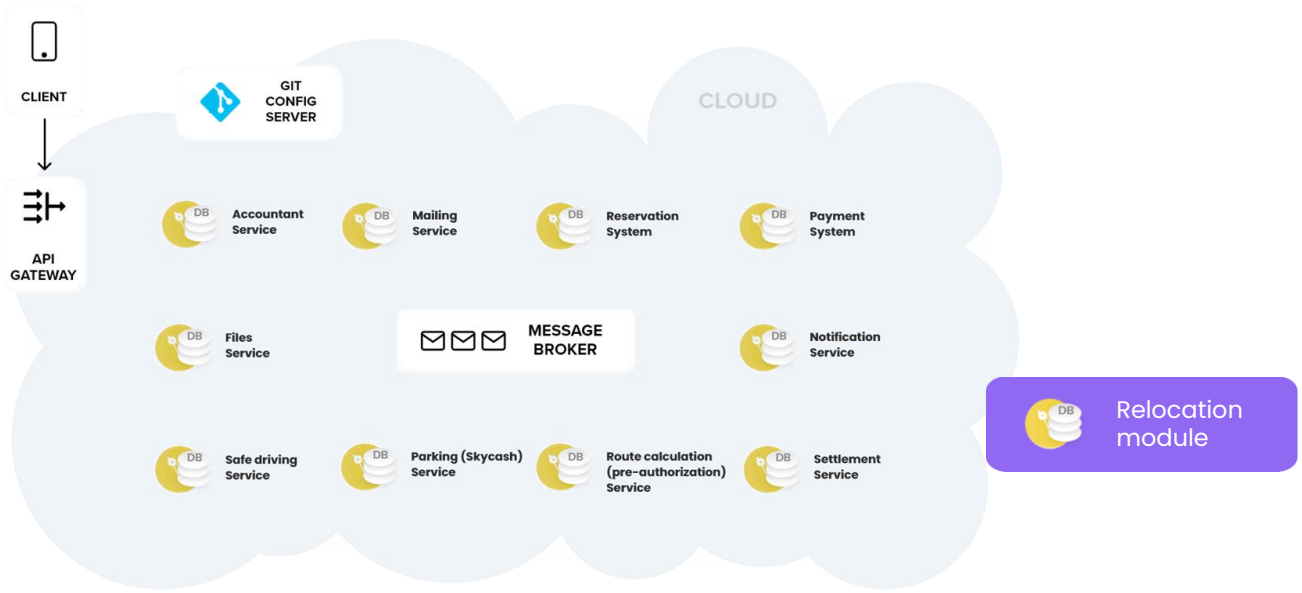
AGH

EUROPEAN UNION
European Regional Development Fund

core•logic

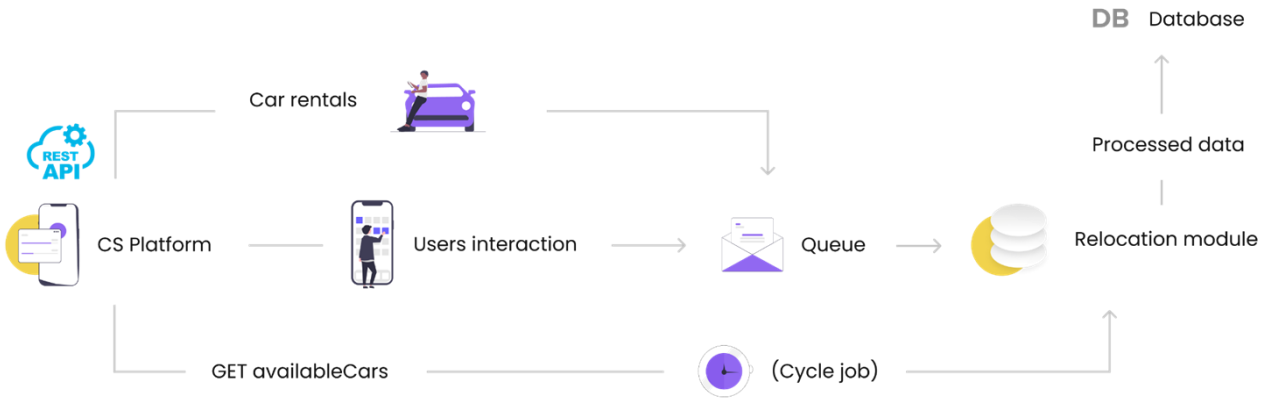
Jak wdrożyć AI w Car Sharing

Architektura platformy car-sharing

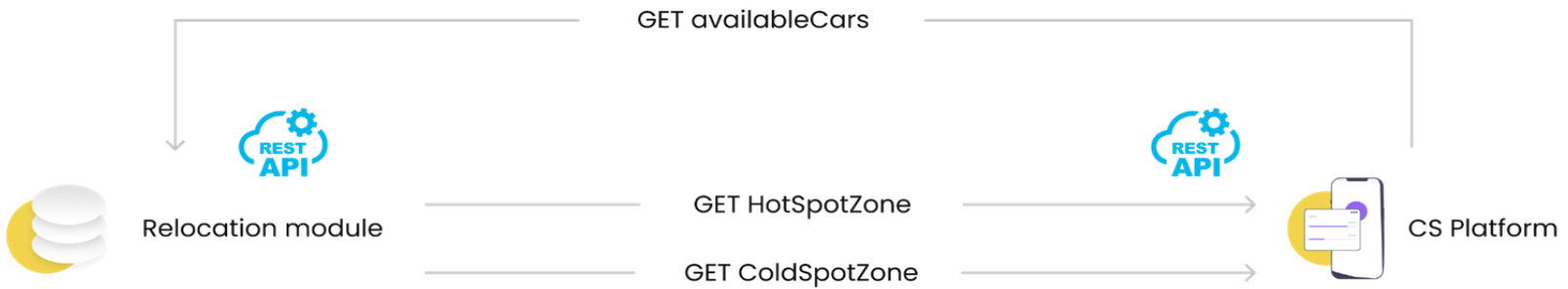


core • logic

Architektura platformy car-sharing



Jak to działa? Dane źródłowe i analiza



Parameters

- TIME range
- Location and range

core•logic

Jak działa moduł od strony technicznej?

Gromadzenie danych

Przetwarzanie danych

Traficar API

API dostępności samochodów

Usługa gromadzenia danych

Gromadzenie historycznych/aktualnych danych dotyczących dostępności samochodów i aktywności użytkowników

Przechowywanie

Rozproszone obliczenia Zbioru Danych Spark

Rozproszone środowisko do przetwarzania dużych zbiorów danych

Aplikacja Python do przetwarzania danych z obsługą Spark

Wstępne przetwarzanie zebranych danych o dostępności samochodów i aktywności użytkowników: grupowanie według lokalizacji, obliczanie początkowych wartości podaży/popytu

Zaplanowane zadanie CornJob Kubernetes

Zaplanowane zadanie z 5-minutowym okresem

core • logic

Jak można wykorzystać wyniki działania modułu AI?

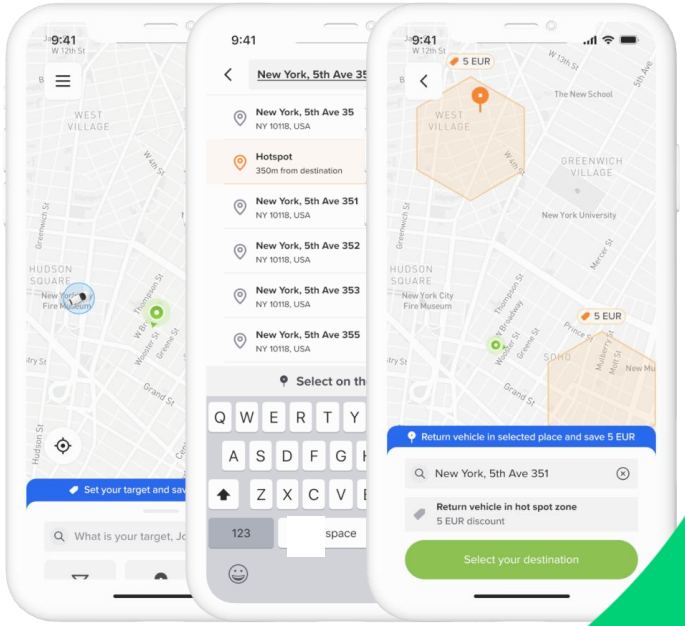
Idea (vide Uber) Podanie celu podróży

Korzyści dla operatora:

- budowa bazy danych
- nowe możliwości biznesowe (hot spot)
- lepsza utylizacja floty
- bonusy uzależnione od stref

Korzyści dla użytkownika:

- transparentność
- lepszy UX
- lepsza oferta -> oszczędność
- Lepsza dostępność usługi



core•logic

Problem: Cold spot

Profit dla operatora carsharing:

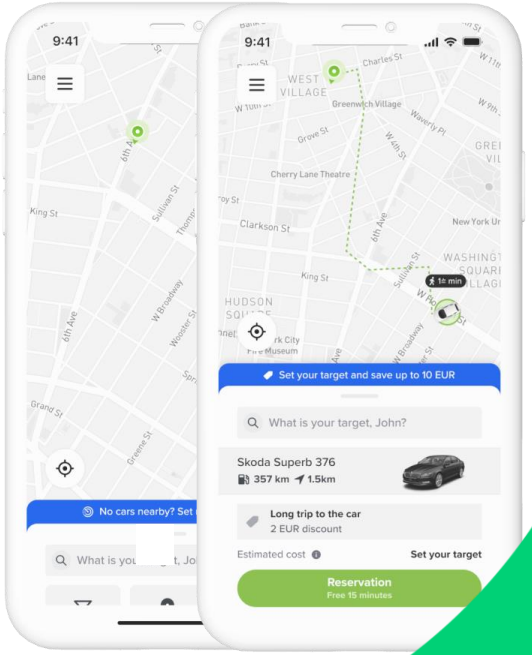
- lepsza utylizacja floty
- Budowa więzi z użytkownikiem, przytrzymanie go w aplikacji

Korzyści dla użytkownika:

- Więcej możliwości (nawet darmowe podróżowanie)
- Oszczędność pieniędzy

*Model predykcyjny

- system wie czy/gdzie będzie zapotrzebowanie na usługę



core•logic

Część II

Motywacje

- **Znane rozwiązania problemu relokacji dla usługi korzystającej ze stacji parkowania oparte na MIP (Mixed Integer Programming)**
- **Granice optymalizacji**

  Wyznaczenie teoretyczną granicę poprawy funkcji celu (wskaźnika wykorzystania pojazdów) rozwiązując problem metodą dokładną dla danych empirycznych
- **Podział na sześcioboki o boku 500m jest arbitralny**

  Może pokrywać obszary, które są odległe w sensie odległości drogowej
- **Liczba sześcioboków jest zbyt duża dla MIP (około 250)**

  W literaturze 15-20 przy czasie optymalizacji sięgającym 6 godzin
- **Dobór wielkości obszaru**

  Osiągnięto bardzo dobre wyniki predykcji liczby dostępnych pojazdów i interakcji użytkowników dla dużych obszarów (dzielnic), ale one spadają wraz ze zmniejszeniem obszaru

Założenia

Podział obszaru świadczenia usługi na kilkadziesiąt stref charakteryzujące się podobnymi własnościami

- Bliską odległością drogową lokalizacji wewnątrz strefy
- Podobnym obserwowanym zachowaniem
- Preferowany jest kulisty kształt
- Ograniczenia dotyczące rozmiarów

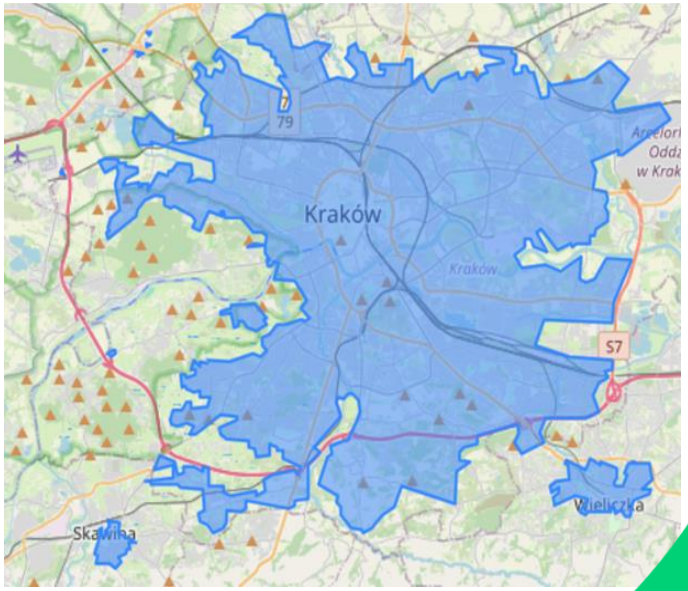
Optymalizacja dwuetapowa

- Pierwszy etap: przemieszczenie pojazdów pomiędzy strefami
- Drugi etap: wybór lokalizacji wewnątrz strefy

Sugerowane rozwiązanie problemu free-floating:
Weikl, S., & Bogenberger, K. (2015). A practice-ready relocation model for free-floating carsharing systems with electric vehicles–Mesoscopic approach and field trial results. Transportation Research Part C: Emerging Technologies, 57, 206-223.

Dane luty 2021 – luty 2022

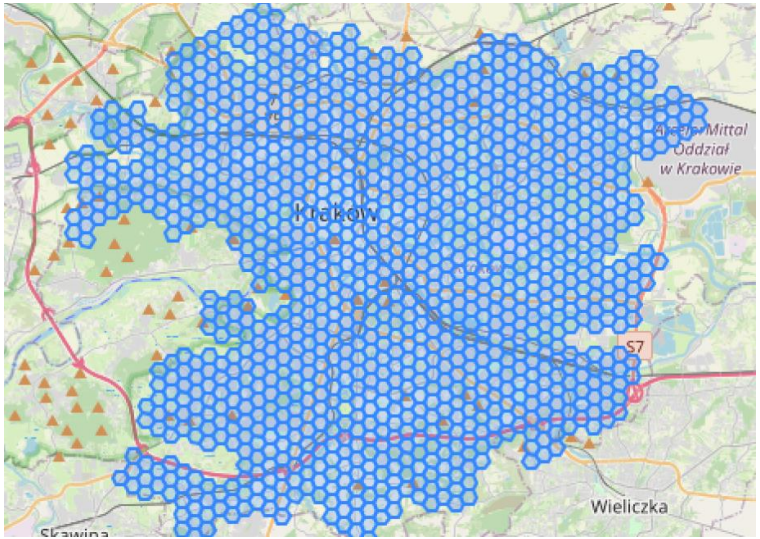
- Obszar oferowania usługi (wyłączono Skawinę i Wieliczkę)
- Przejazdy: 239 tys.
- Postoje: 393 tys.
- Interakcje użytkowników z systemem: 5.8 mln



core • logic

Podział na strefy

- Obszar oferowania usługi pokryty sześciobokami o boku 250m
- Zaimplementowano aglomeracyjny algorytm grupowania
- Dodatkowe opcje:
  - Ograniczenie rozmiaru grupy
  - Wchłanianie wysp
  - Wchłaniania pojedynczych sześcioboków na granicy obszarów



core•logic

Składniki odległości

- **Rzeczywista odległość w sieci drogowej pomiędzy centroidami**
- **Gęstość sieci drogowej**

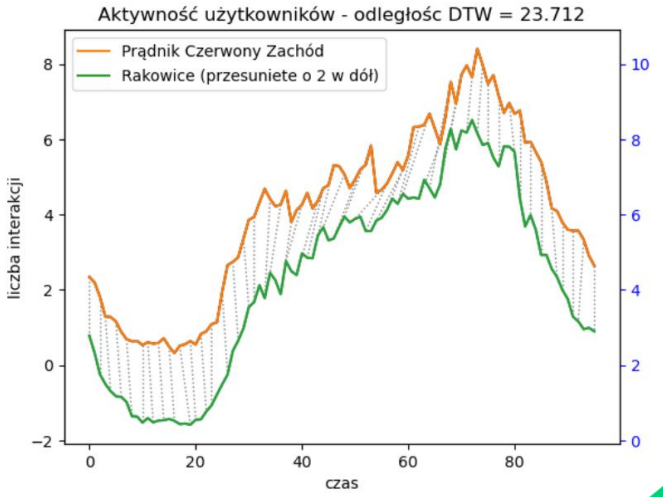
  Wyznaczona z danych OSM
- **Kształt potencjalnie tworzonej nowej grupy**

  Preferowane kształty kuliste
- **Podobne wzorce czasowe**
  - liczby zaparkowanych pojazdów
  - interakcji użytkowników

  Odległość DTW (Dynamic Time Warp)
- **Zastosowane wagi**
- **Walidacja grupowania**
  - Jakość predykcji
  - Osiągnięto średnią wartość:

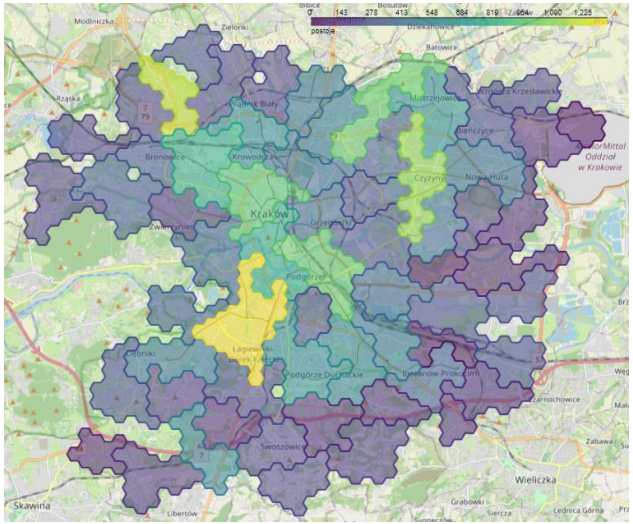
    r2=0.91 dla predykcji w horyzoncie 30 min i

    r2=0.84 dla horyzontu 60 min.
- **Wyznaczono 63 strefy**

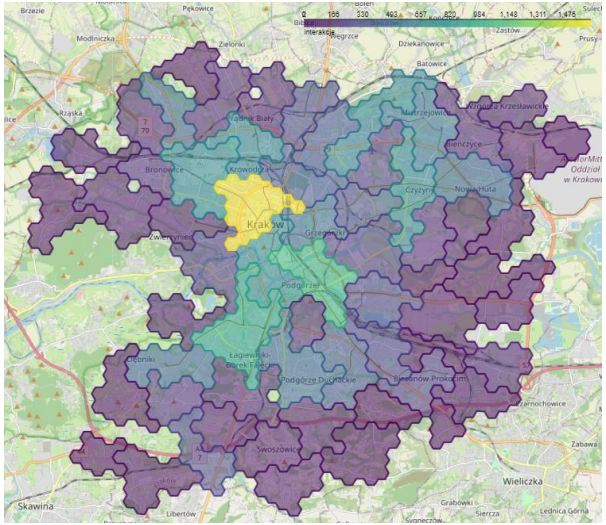


Analiza danych stref

Postoje w ciągu doby: 7.4-1334



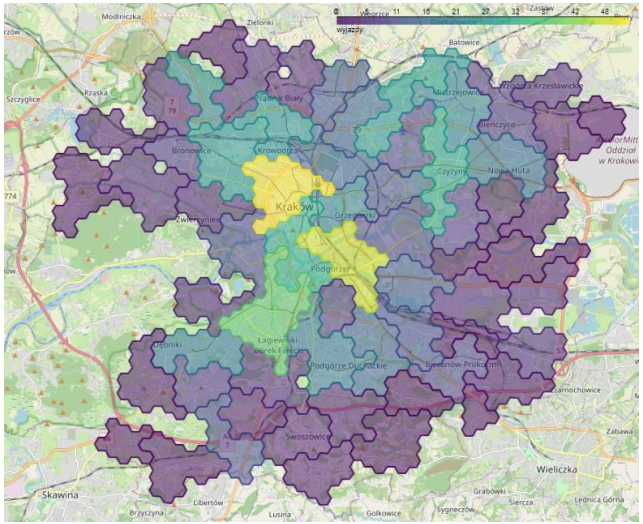
Interakcje w ciągu doby: 2.4-1607



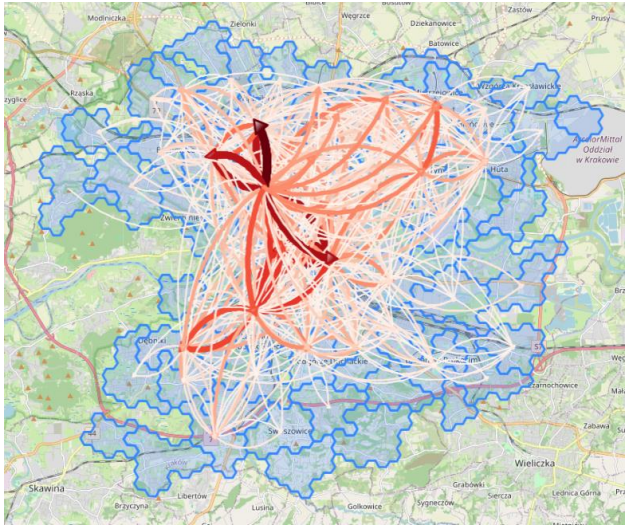
core•logic

Analiza danych stref

Wyjazdy w ciągu doby: 0.2-51



Przepływy między strefami: 0-4626



core•logic

Optymalizacja

- Typ modelu: Stochastic MIP (Mixed Integer Programming)
- Czas dyskretyzowany do 15 min, horyzont 24h
- Parametry modelu:
  - $D[i,j,t]$ - znane zapotrzebowanie klientów na przejazd ze strefy i do j w momencie t
  - $T[i,j,t]$ – czasy przejazdu pomiędzy strefami i oraz j rozpoczętego w momencie t
  - Inne: początkowa liczba pojazdów w strefach, liczba pojazdów, liczba pracowników
- Zmienne decyzyjne:
  - $x[i,j,t]$ – liczba przydzielonych klientom pojazdów do realizacji przejazdu pomiędzy strefami i oraz j w momencie t
  - $y[i,j,t]$ – liczba przydzielonych pracownikom pojazdów do realizacji przemieszczenia pomiędzy strefami i oraz j w momencie t

Model optymalizacyjny

Ograniczenia

- $x[i,j,t] \leq D[i,j,t]$ -liczba przydzielonych pojazdów nie przekracza zapotrzebowania
- Liczba pojazdów w strefach wyznaczana na podstawie przyjazdów i odjazdów
- Przydział pojazdu następuje co 15 minut
- Czas przejazdu na podstawie danych historycznych
- Liczba pracowników na stacjach zależy od przemieszczanych pojazdów

Funkcja celu f – sumaryczny czas przejazdów zrealizowanych przez klientów

Przebieg optymalizacji

- Zakładamy, że znane są parametry rozkładu Poissona $\Lambda[i,j,t]$ pojawienia się zapotrzebowania na przejazd. Na jego podstawie losowana jest pełna macierz zapotrzebowania $D[i,j,t]$
- Symulacja podjętych decyzji i wykonanych przejazdów tworzy **scenariusz** dla danego zapotrzebowania $D[i,j,t]$
- Wykonywana jest pewna liczba scenariuszy (n=100).
- Wartość średnia funkcji celu przybliża wartość oczekiwaną dla wszystkich możliwych scenariuszy

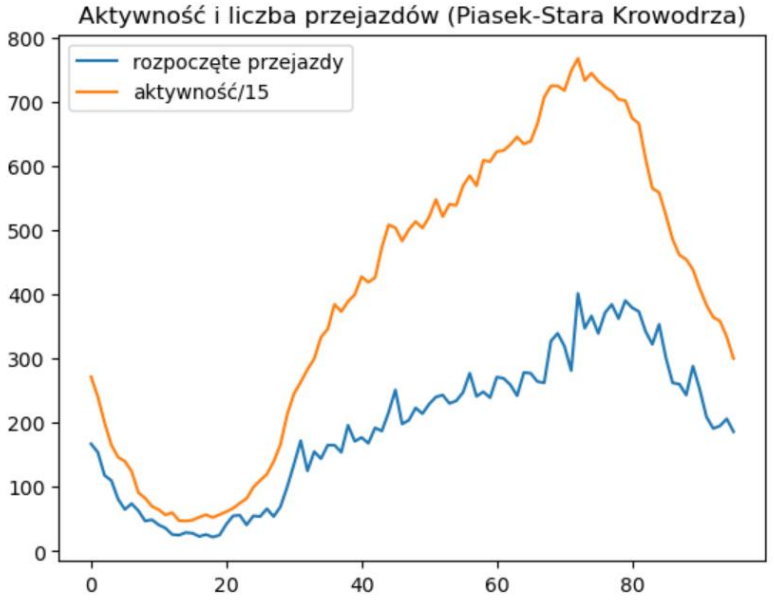
$$\sum_{i=1}^{n} f(s_i) \approx \boldsymbol{E}_S(f)$$

- Optymalizacja jest dwupoziomowa – na pierwszym poziomie ustalane są wartości zmiennych decyzyjnych pierwszego etapu (np.: liczba pracowników), a na drugim aproksymowana wartość oczekiwana funkcji celu

Skąd parametry rozkładu Λ?

Założenia

- Średnia wartość $\lambda[i,t]$ zapotrzebowania na przejazdy ze strefy i w momencie t zależy od aktywności użytkowników
- Wartość ta rozkłada się na $\Lambda[i,j,t]$ proporcjonalnie do obserwowanej liczby przejazdów miedzy strefami (dane historyczne)

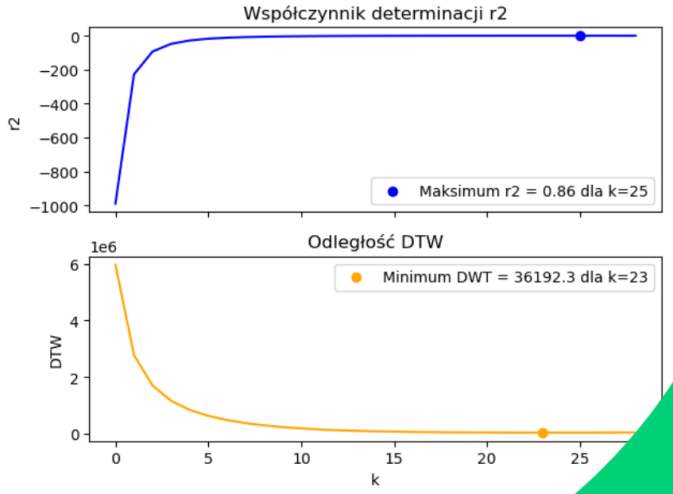


core•logic

Współczynnik skalujący: zapotrzebowanie = interakcje/k

Założenia:

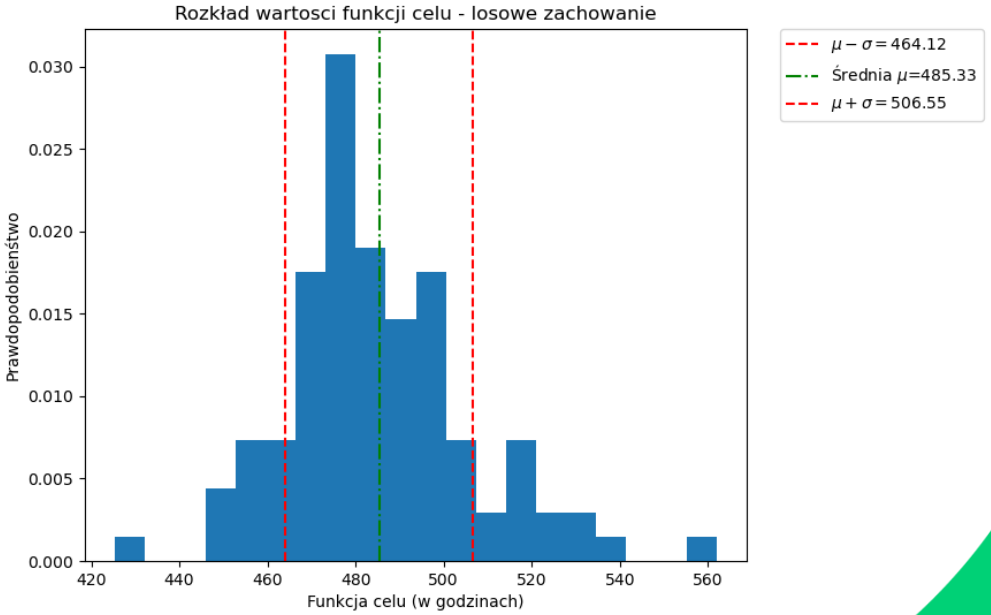
- Na rysunku porównanie odległości pomiędzy przeskalowaną krzywą dobową interakcji i wykonanymi przejazdami
- Najmniejsza odległość dla k w przybliżeniu 23-25
- Przebiegi są wtedy bliskie pokrycia.
- Czasem aktywność byłaby mniejsza niż liczba rozpoczętych przejazdów…
- Przyjęto k=15



core•logic

Wyniki: losowe zachowanie – poziom odniesienia

- Losowy przydział pojazdów
- Każdy scenariusz powtórzony 20-krotnie

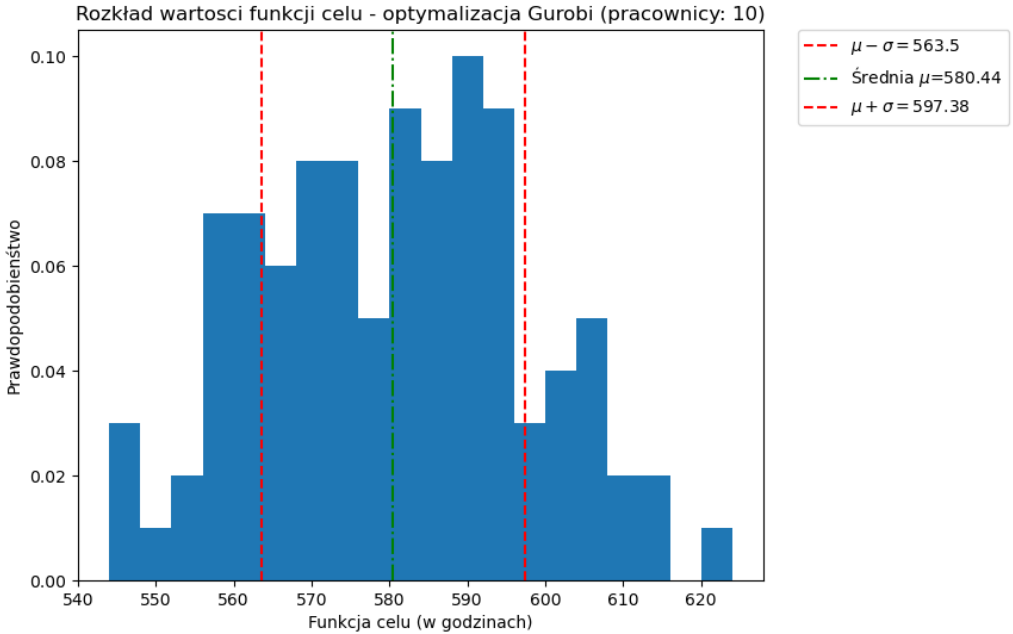


core•logic

Wyniki: optymalizacja w stochastycznym modelu MLP

- Ten sam zestaw scenariuszy
- Solwer: Gurobi
- 100 scenariuszy
- Czas: około 1 min/scenariusz

Poprawa o około 20%



core•logic

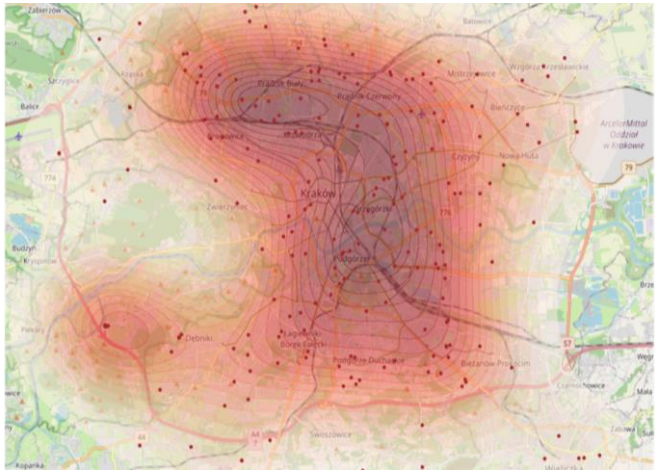
Estymacja jądrowa

Dany jest zbiór punktów $X = \{x_1, \dots, x_n\}$

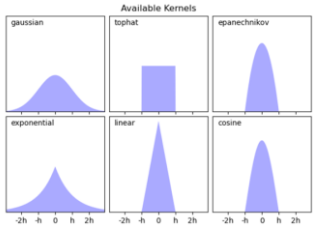
Gęstość prawdopodobieństwa określona jest wzorem:

$$f(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{d(x,x_i)}{h}\right)$$

$d(x, x_i)$- odległość
K – funkcja jądra
h – szerokość pasma



Łatwa do wyznaczenia w 2D dla odległości euklidesowej



core•logic

Estymacja jądrowa w sieci drogowej

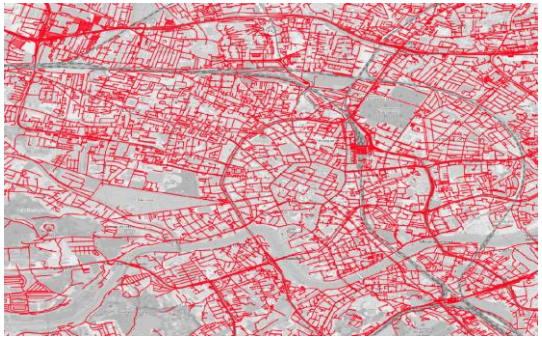
- Zastąpienie odległości euklidesowej odległością drogową
- Analogia do hot-spotów, czy pikseli dla KDE ale obiektami agregującymi są odcinki dróg
- Rozkład prawdopodobieństwa dla postoju pojazdów lub interakcji użytkownika

Zielone punkty – zaparkowane pojazdy
Szarym kolorem oznaczono ulice z przypisaną wartością rozkładu powyżej wybranego progu

core•logic

Estymacja jądrowa w sieci drogowej

- Wykorzystano mapę OSM (Postgres z roszerzeniem PostGIS)
- Przetwarzanie wstępne
  - Podział sieci drogowej na odcinki (do 60m)
  - Redukcja liczby odcinków (wyłączenie dróg z zakazem parkowania, dróg określonej kategorii, rond, itp.)



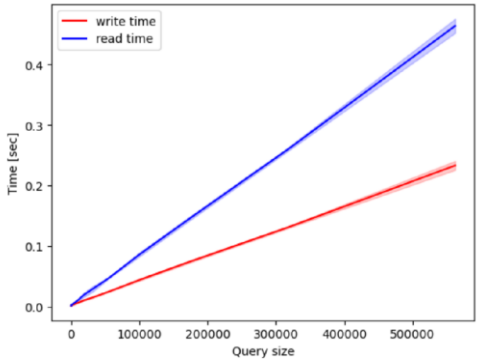
core • logic

Problem wydajności

- Liczba odcinków po redukcji – około 100 tys
- Formuła gęstości $f(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{d(x,x_i)}{h}\right)$
  wymaga dla danej obserwacji x_i wyznaczenia 100 tys. wartości odległości dla par (x, x_i)
- Wewnętrznie wykorzystywana jest funkcja pgr_dijkstra() z rozszerzenia pgRouting dla Postgres

Usprawnienia:

- obliczenia tylko dla odcinków odległych o dwie szerokości pasma
- Zastosowanie mechanizmu pamięci podręcznej:
  Tablica 100_000 x 100_000 odległości drogowych przechowywana jest w pliku o rozmiarach około 37GB
- Przyspieszenie – około 30 krotne

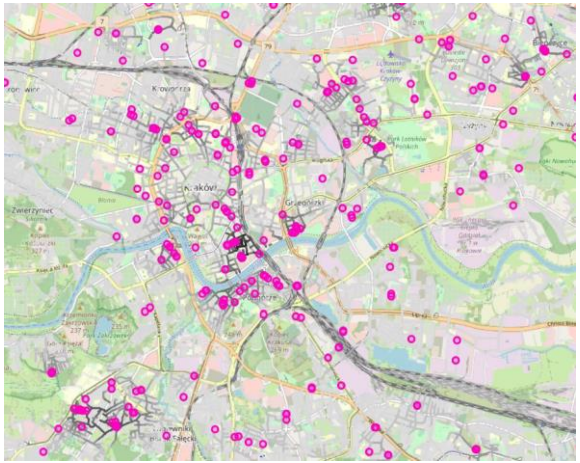


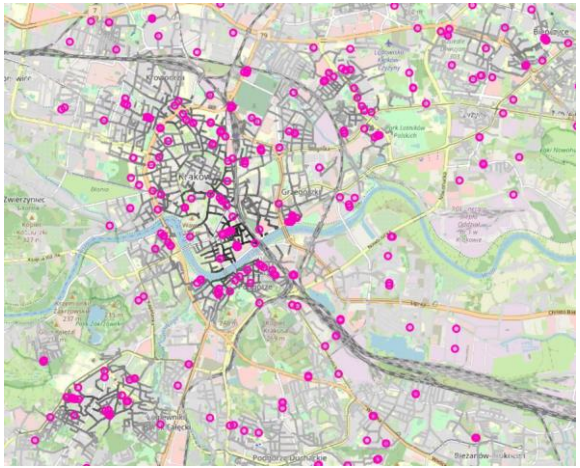
core•logic

Dobór funkcji jądra i pasma

Otrzymane rozkłady zależą od funkcji jądra i szerokości pasma.

Po lewej rozkład dla szerokości pasma $h = 500\ m$, po prawej $h = 1\ km$





core • logic

Dobór funkcji jądra i pasma

Walidacja:

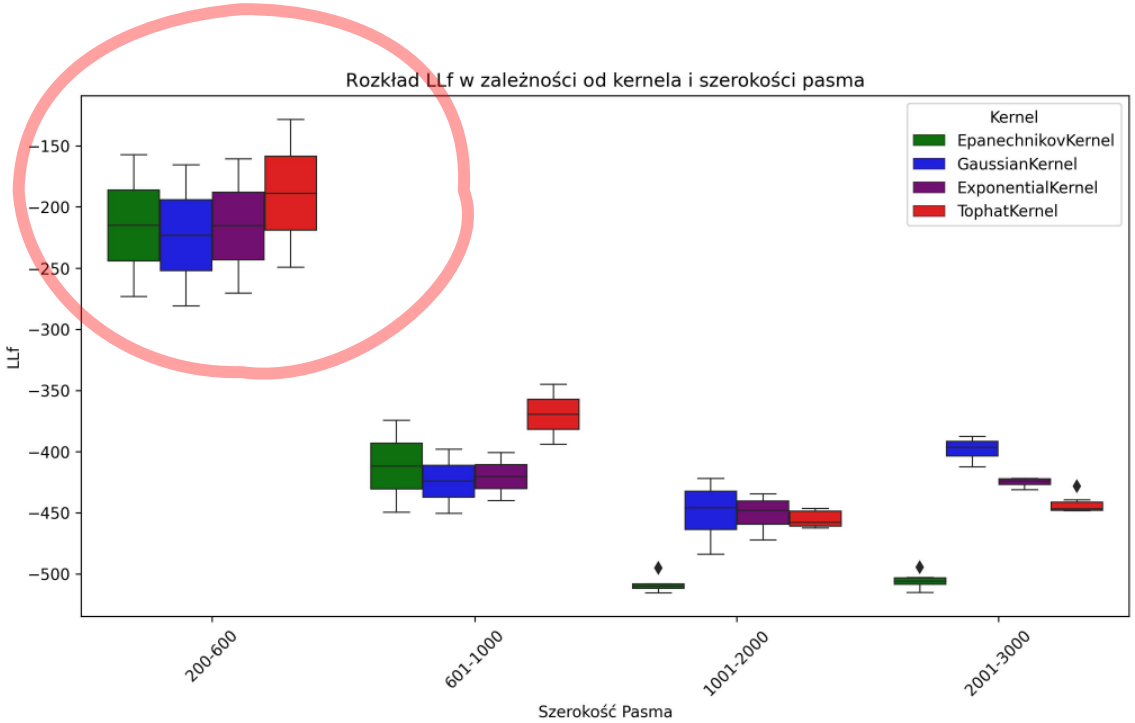
- Podział obserwacji $X = \{x_1, \dots, x_n\}$ na dwa zbiory X_{train} i X_{test}
- Budowa modelu rozkładu $f(x)$ dla obserwacji z X_{train}
- Wyznaczenie prawdopodobieństwa pojawienia się obserwacji z X_{test}

$$P(X_{test}) = \prod_{x_i \in X_{test}} f(x_i)$$

lub po przekształceniu:

$$\mathrm{LL}(X_{test}) = \ln(P(X_{test})) = \sum_{x_i \in X_{test}} \ln(f(x_i))$$

- Wybór parametrów, dla których $LL(X_{test})$ ma najwyższą wartość



core•logic

Propagacja informacji w grafie

Jądrowa estymacja w grafie sieci drogowej może być postrzegana jako jedno z rozwiązań bardziej ogólnego zagadnienia propagacji informacji w grafie, w którym węzły $i=1,\ldots,n$ przechowują swój stan (poziom aktywacji) $x[i]$, uaktualniają go po otrzymaniu komunikatu od sąsiadów i rozsyłają dalej komunikaty

```
Algorytm iteracji dla i-tego węzła
          v = receive()                          # przyjęcie komunikatu
          x[i]<-f(x[i],v)                        # uaktualnienie stanu
          if t(x[i],v) > próg_aktywacji:                  # czy rozsyłać informację
                    for j in neighbors():
                              send(j,g(x[i],v))   # wysłanie komunikatu do sąsiadów
```

Konkretna implementacja zależy od doboru funkcji:

- Uaktualnienie stanu za pomocą `f(x[i],v)` – np. sumowanie
- Decyzja o rozesłaniu informacji `t(x[i],v)` – na podstawie stanu lub otrzymanego komunikatu
- Wybór zbioru sąsiadów `neighbors()` – pojedyncze czy wielokrotne odwiedziny
- Wartość do przesłania: `g(x[i],v)` – np. określona funkcja jądra

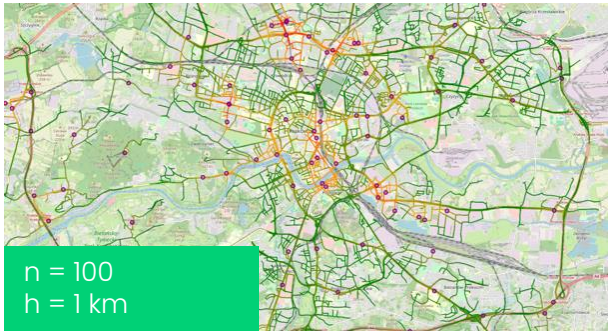
Dobór funkcji jądra i pasma

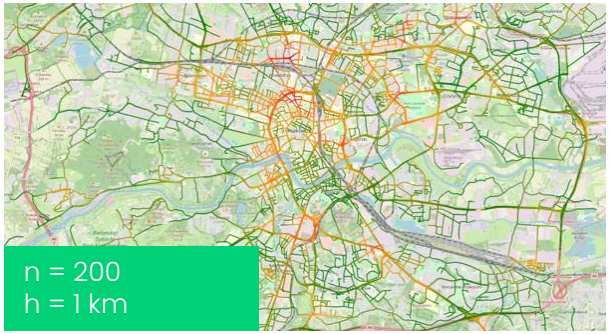
Przebadano następujące algorytmy:

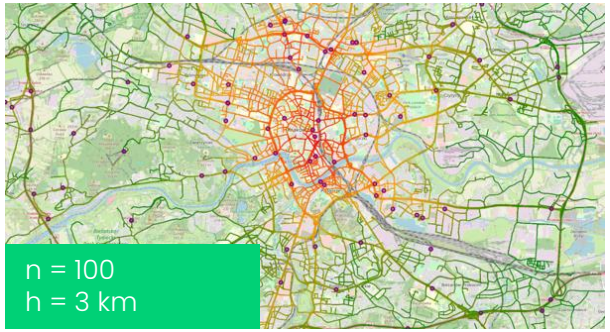
- **Wavefront** – przeszukiwanie grafu wszerz, dla pojedynczego źródła aktywacji węzły odwiedzane tylko raz
- **Pregel** – możliwość wielokrotnego przesyłania komunikatów wstecznych, propagacja następuje w superkrokach, w pierwszej części uaktualnienie stanu na podstawie wszystkich odebranych komunikatów, w drugiej rozesłanie komunikatów
- **RandomWalk** – losowy wybór tylko jednego sąsiada do odwiedzenia. Prawdopodobieństwo wyboru zależne od klasy drogi

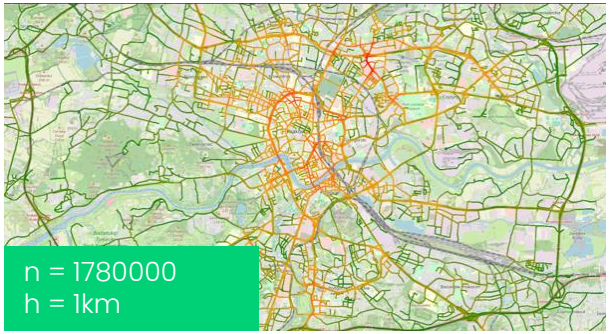
W każdym przypadku zastosowano kernel Gaussa o różnej szerokości pasma.
Eksperymenty przeprowadzono dla przypadków różniących się liczbą aktywowanych punktów początkowych

Wavefront

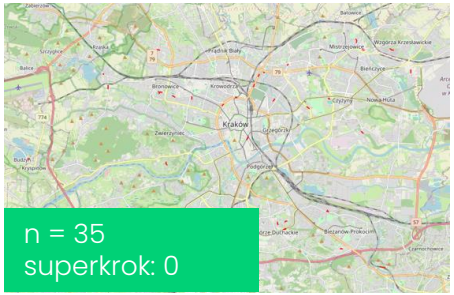


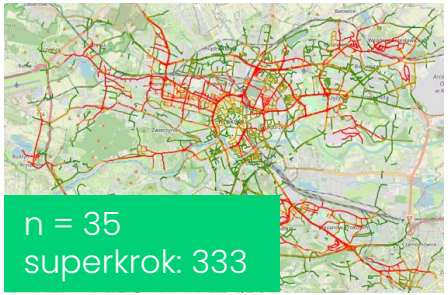


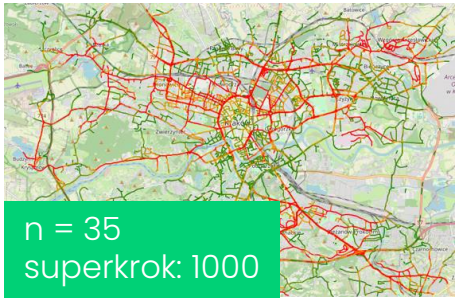


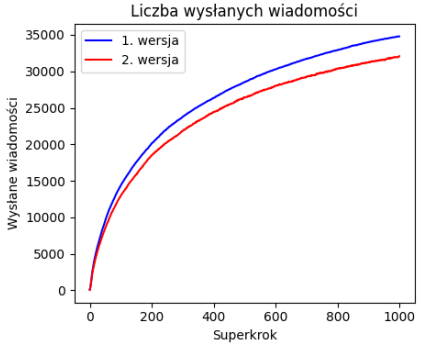


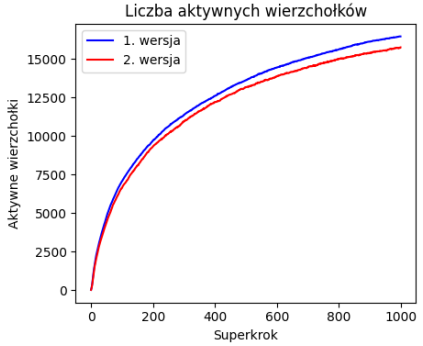
Pregel

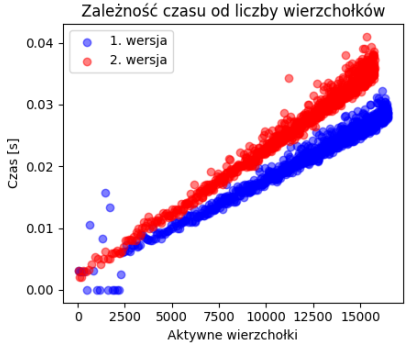






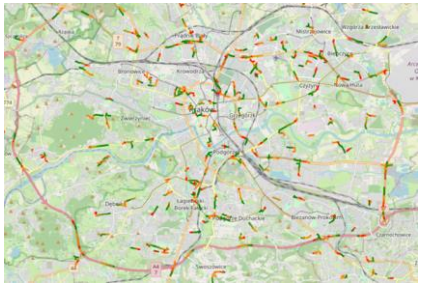




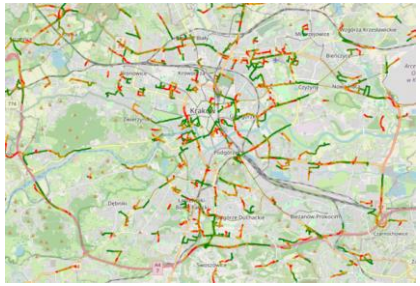


core•logic

Random Walk



n = 100
h = 500 m



n = 100
h = 2 km

n = 100
h = 16 km

n = 100
h = 64 km

core • logic

Algorytmy propagacji

- **Obiecujące ze względu na dużą szybkość działania**
  Możliwa implementacja na GPU
- **Porównanie wynikowych grafów – odległość wektorów aktywacji**
  Potrzeba opracowania bardziej semantycznej miary
- **Przetestowane na danych syntetycznych**
  Potencjalne zastosowanie do estymacji prawdopodobieństwa wymaga walidacji, jak dla estymacji jądrowej

Prace

Przedstawione badania zrealizowano w ramach następujących prac dyplomowych:

- Kamil Roman:
  Grupowanie i predykcja dla danych komunikacyjnych z systemu Car Sharing, 2023, praca magisterska
- Konrad Podgórski:
  Estymacja gęstości prawdopodobieństwa dla danych ruchu drogowego, 2023, praca magisterska
- Dominik Kikla:
  Optymalizacja rozmieszczenia pojazdów w systemie Car Sharing, 2024, praca inżynierska
- Patryk Gęgotek:
  Algorytmy propagacji informacji w dużych grafach, 2024, praca inżynierska

Podsumowanie

- Platforma Core Logic oferuje **dedykowane oprogramowanie dla usługi Car Sharing**
- Przedstawiono koncepcję zastosowania sztucznej inteligencji dla poprawy efektywności usługi oraz produkcyjne rozwiązania techniczne
- Badania przeprowadzone przy okazji wdrożenia są kontynuowane:
  - podział na strefy
  - algorytmy optymalizacji
  - metody szacowania popytu i podaży oparte na ich estymacji dla grafu sieci drogowej