Projekt : Clustering

H.Drążkowski, M.Wilk

08 czerwiec 2021

Wydział Matematyki i Nauk Informacyjnych
Politechnika Warszawska

Agenda

Agenda

1. Agenda
2. EDA
3. Feature Engeneering
4. Modelowanie
5. Wyniki

Agenda

Pierwszym celem projektu jest zastosowanie metod analizy grupowania danych (ang. clustering).
Drugim celem projektu jest odpowiedź na pytanie jak podobne są do siebie starożytne teksty religijne.

EDA

EDA

Inspiracją do stworzenia danych wyjściowych był "Project Gutenberg". Czyszczenie danych przeprowadzili PREETI SAH i ERNEST FOKOUE w (2019) "WHAT DO ASIAN RELIGIONS HAVE IN COMMON? AN UNSUPERVISED TEXT ANALYTICS EXPLORATION".
Dostępne były dwa źródła danych:

- Macierz rzadka gdzie zmiennymi byla liczba wystapien danego slowa w danym fragmencie
- Orginalny tekst ze wszystkimi fragemntami skonkatenowanymi

EDA

Fragmenty pochodzą z 8 tekstów z czterech religii. Stąd potencjalnie rodzą się dwie warstwy etykietek

- Hinduism : Yogasutras, Upanishads
- Buddishm : Four Noble Truth of Buddhism
- Taoism : Tao Te Ching
- Christianity : Book of Proverb, Book of Ecclesiastes, Book of Ecclesiasticus, Book of Wisdom

EDA

Wnioski po eksploracyjnej analzie danych macierzy zliczeń słów:

- Jest bardzo dużo słów, które rzadko występują, ponad 50% z nich występuje 2 razy lub rzadziej.
- Mamy do czynienia z problemem dużej ilosci wymiarów. Obserwacji jest 589, a zmiennych 8266
- Niektóre słowa pojawiają się często, rzadko zdażają się słowa, które występują wiele ilości razy we fragmencie
- Fragmenty różnią się iloscią zawieranych słów. Zdecydowana większość jest w przedziale 0-200. Potem jest kilkadziesiąt do 400, powyżej 400 jest niewiele obserwacji.

Feature Engeneering

Feature Engeneering

Po przeczyszczeniu orginalnego tekstu tak aby pozbyć się dziwnych znaków i uzyskać mniej więcj orginalny podział (z dokładnością do pewnych słów nieinformatywnych usuniętych przez autorów artykułu) tworzymy nową ramkę danych.

Feature Engeneering

Dodajemy

- Polarity (użycie większości słów nacechowanych negatywnie albo pozytywnie)
- Subjcetivity (użycie słów uważanych za opiniotwórcze)
- FRE (indeks mierzący skomplikowanie tekstu)
- ARI (indeks mierzący skomplikowanie tekstu)
- avg_wordlength (długość średnia słowa we fragmencie)

Feature Engeneering

Na orginalnej macierzy dokonaliśmy kilku przekształceń

- Aby zmniejszyć liczbę wymiarów usuwamy słowa ktre wystąpiły co najywżej dwa razy w całym korpusie. (zostało 4394)
- Aby jakoś zachować informacje mierzymy ile razy wystąpiło jakiekolwiek unikatowe (<2) słowo w danym fragemncie. Do tego dodajemy ile takich unikatowych słów miał fragment.
- Na tym robimy PCA, uzyskując 91% tłumaczonej wariancji zostajemy przy 200 składowych głónych

Feature Engeneering

Ostatecznie przed modelowaniem

- Łączymy ze sobą ramke danych po PCA i z orginalnej analizy tekstu za pomocą narzędzi NLP.
- Wystandaryzowujemy niektóre zmienne, bo nowo dołączone cechy miały znacznie większe wartości od tych zwróconych przez PCA, tak aby wszystkie kolumny były mierzone w podobnej skali.

Modelowanie

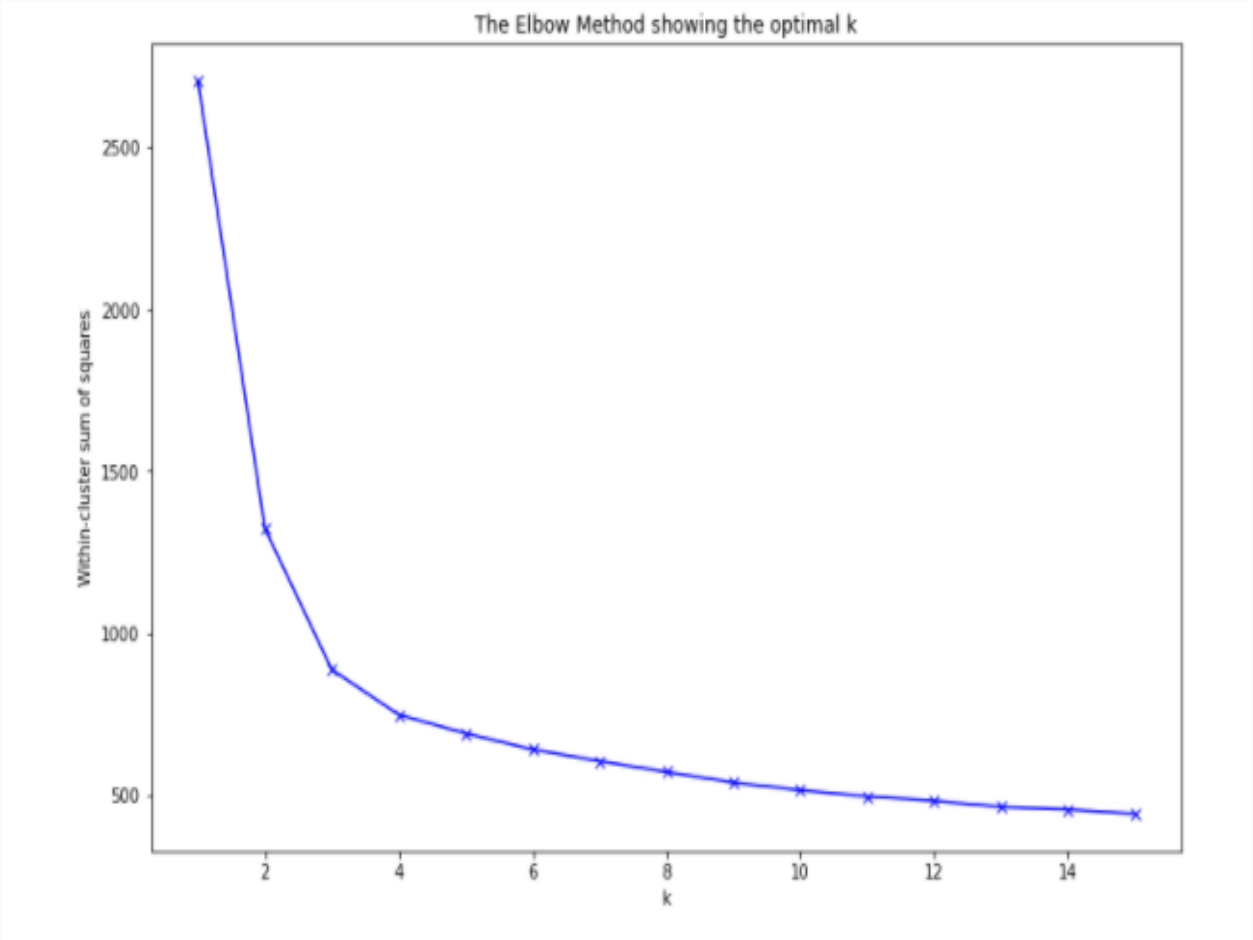
Modelowanie

Patrzyliśmy na beterię metod :

1. Kmedioids
2. Kmeans
3. Agglomerative z połaczneiami
    - Single
    - Average
    - Complete
    - Ward
4. DBSCAN
5. GMM

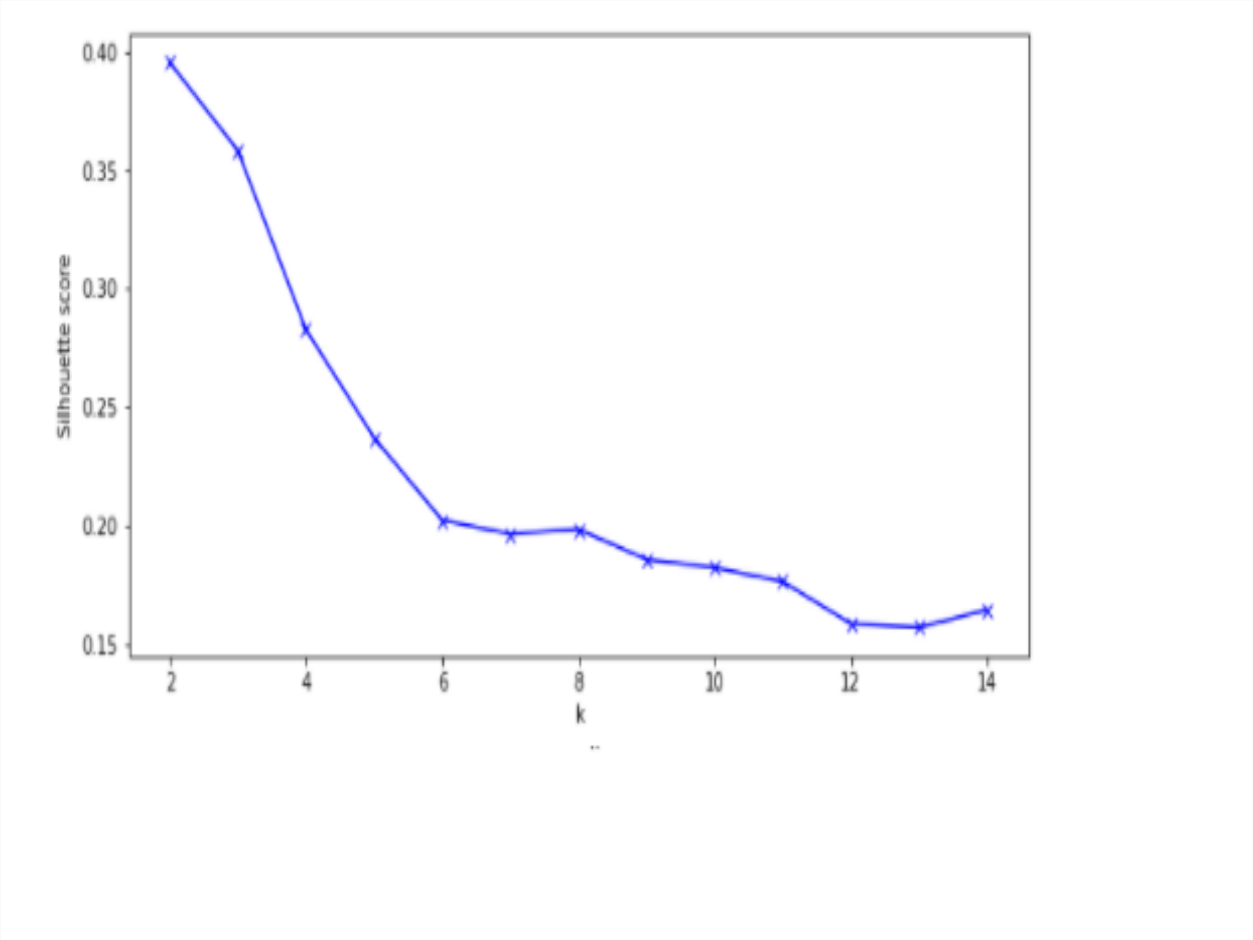
Modelowanie

K-Means



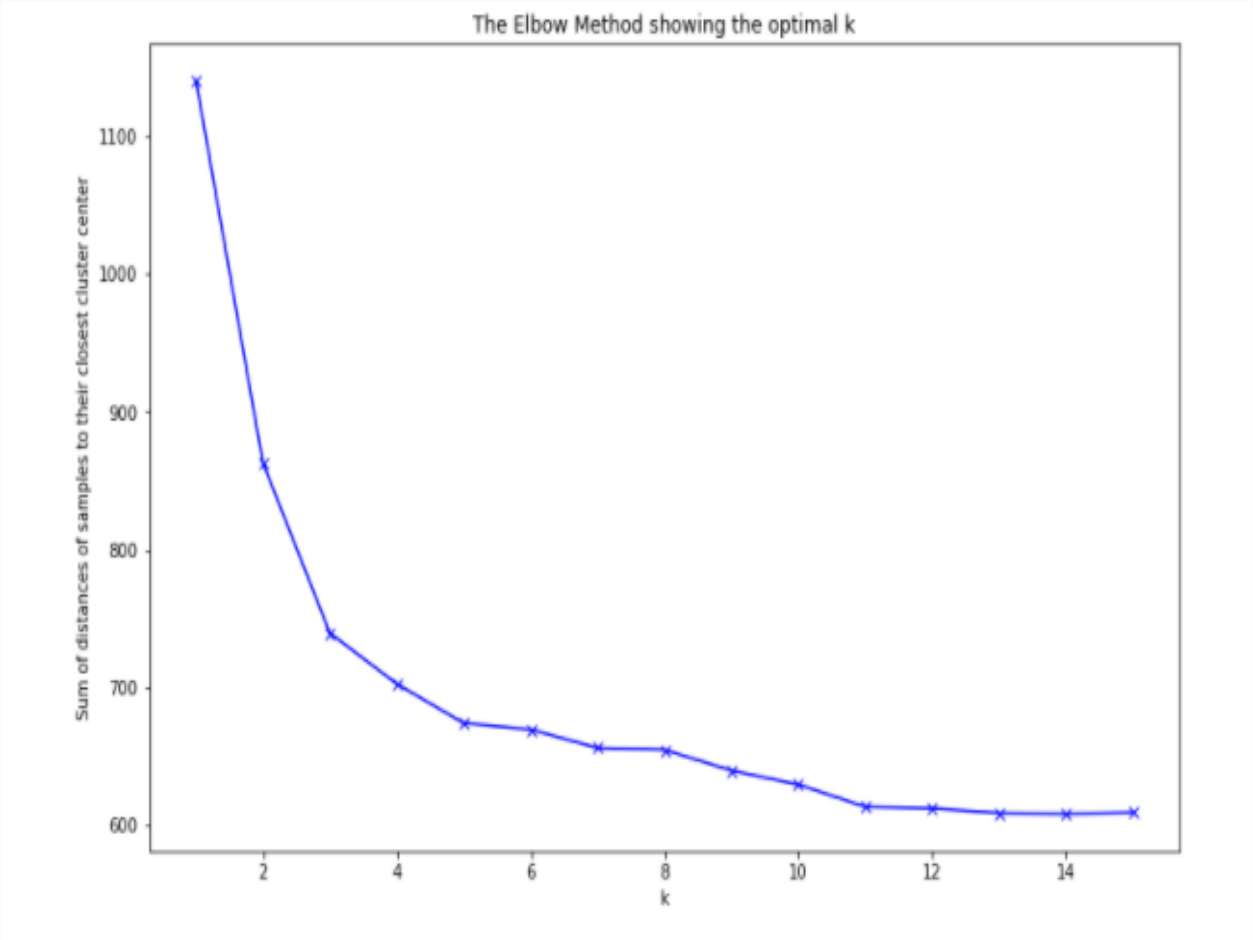
Modelowanie

K-Means



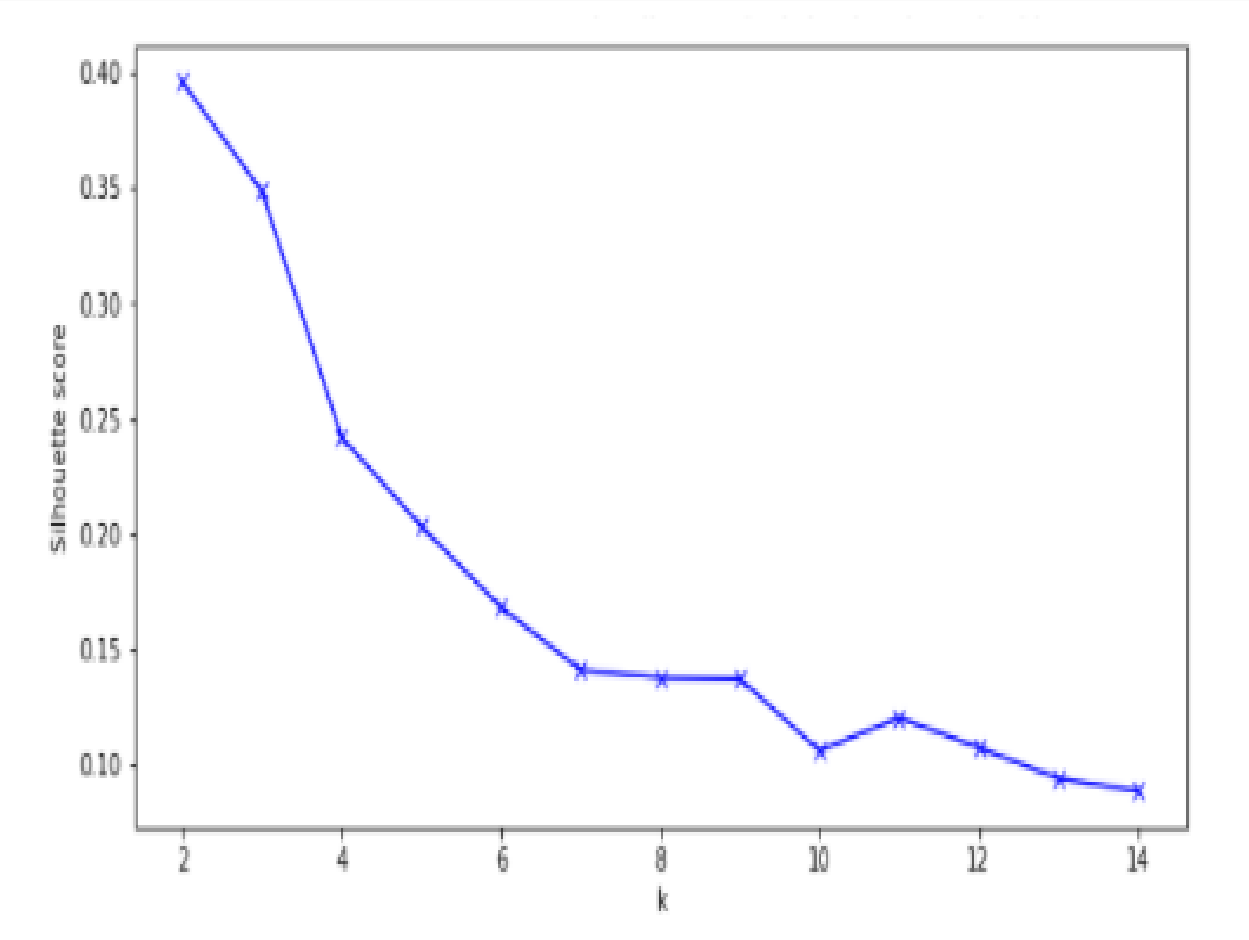
Modelowanie

K-Medioids



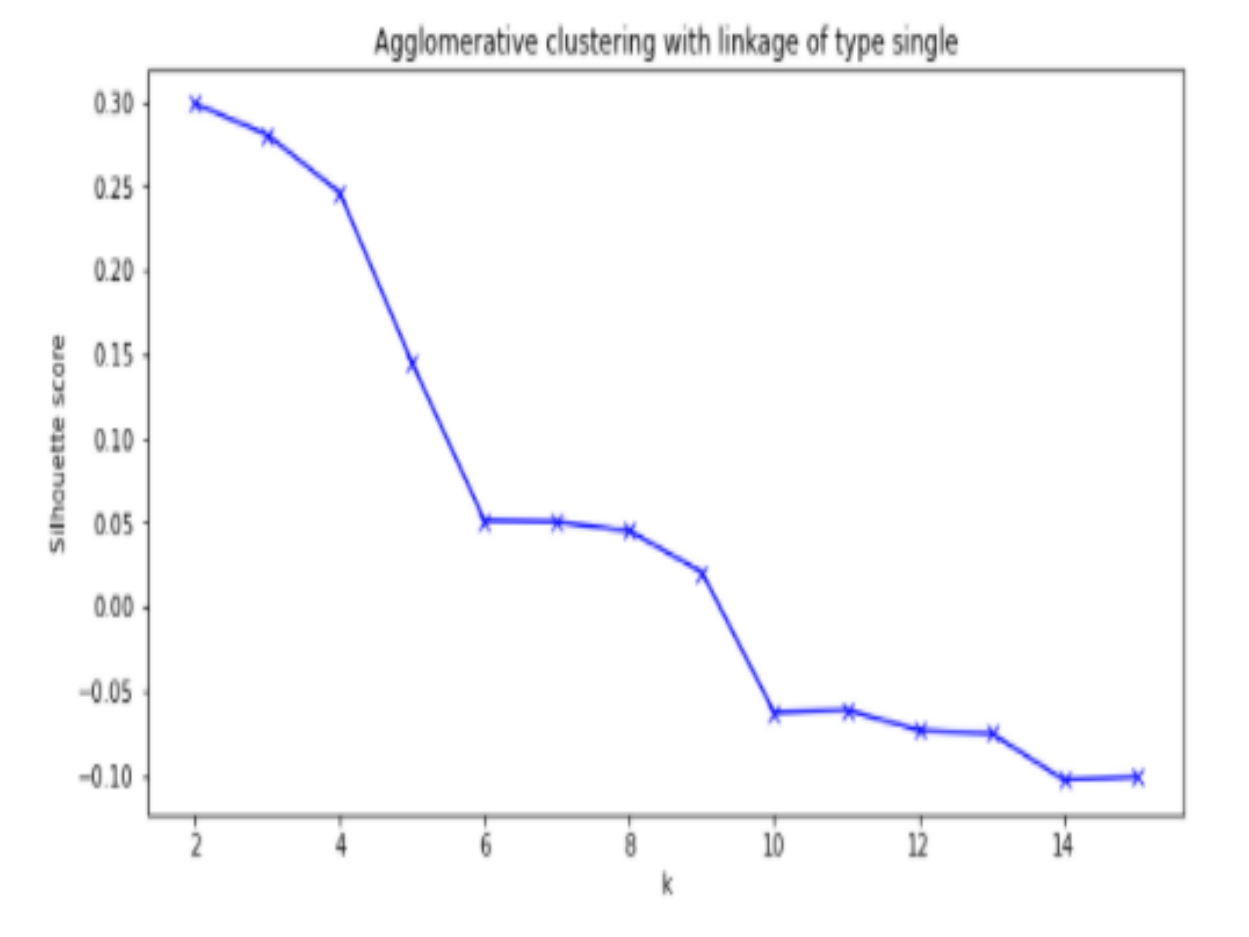
Modelowanie

K-Medioids



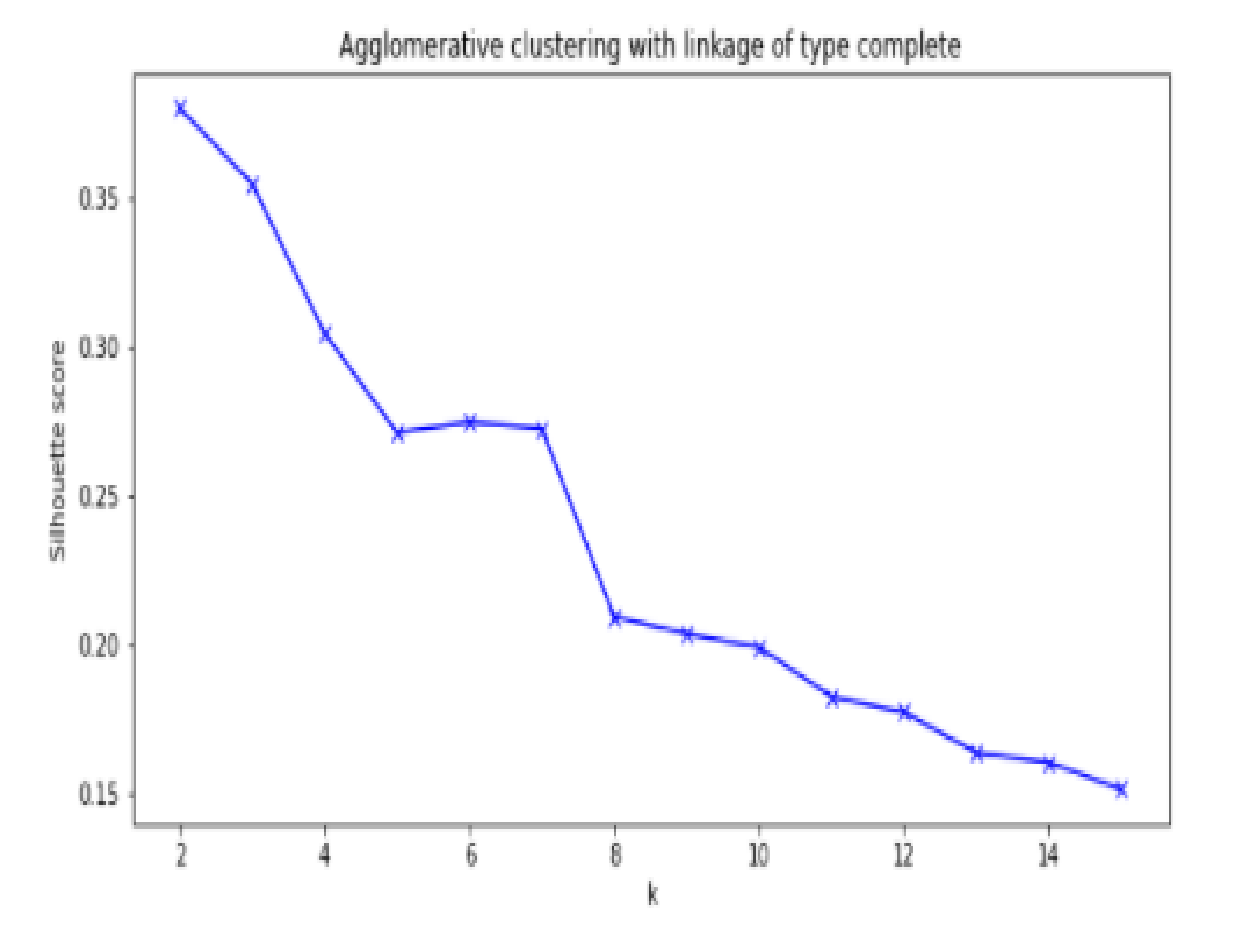
Modelowanie

Agglomerative



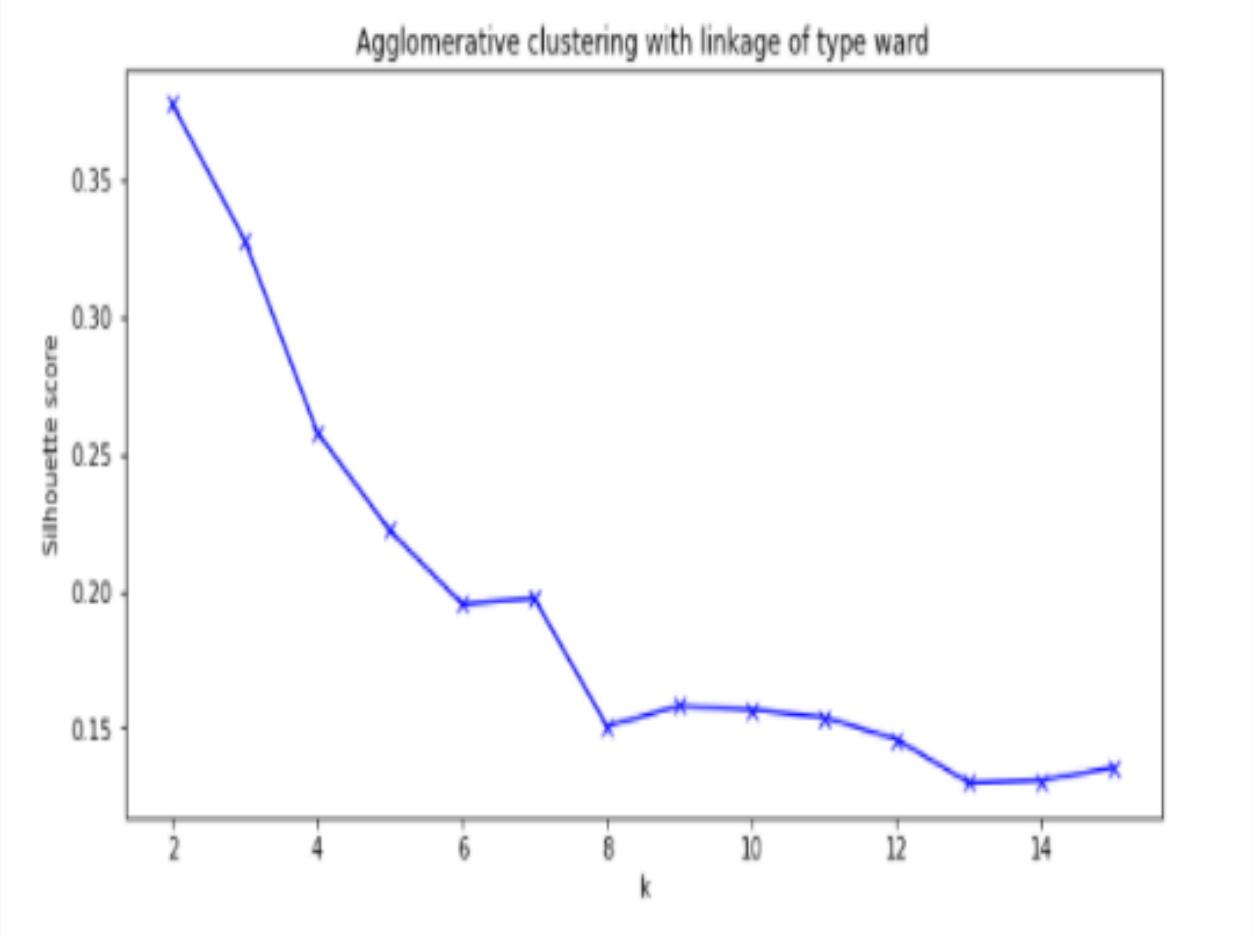
Modelowanie

Agglomerative



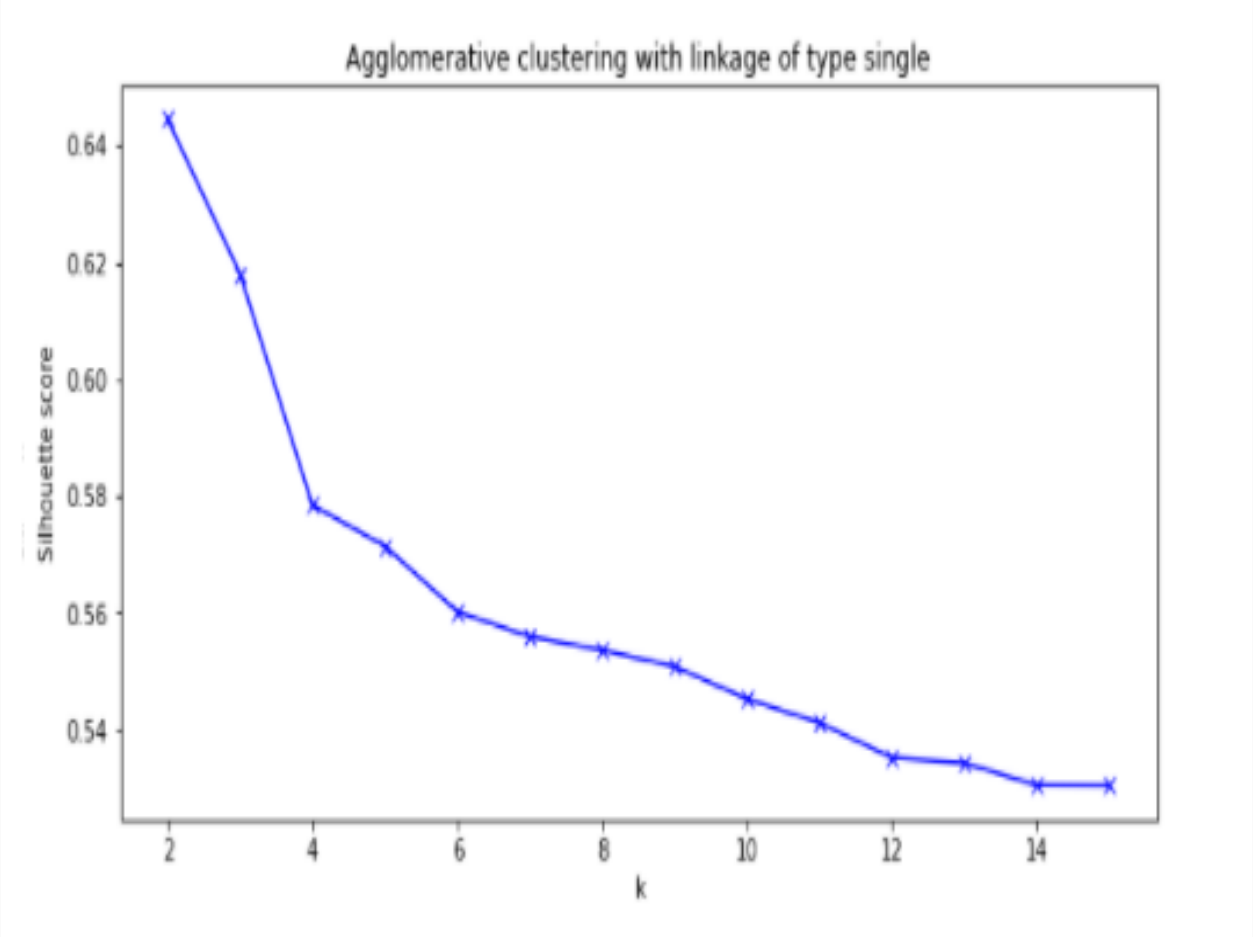
Modelowanie

Agglomerative



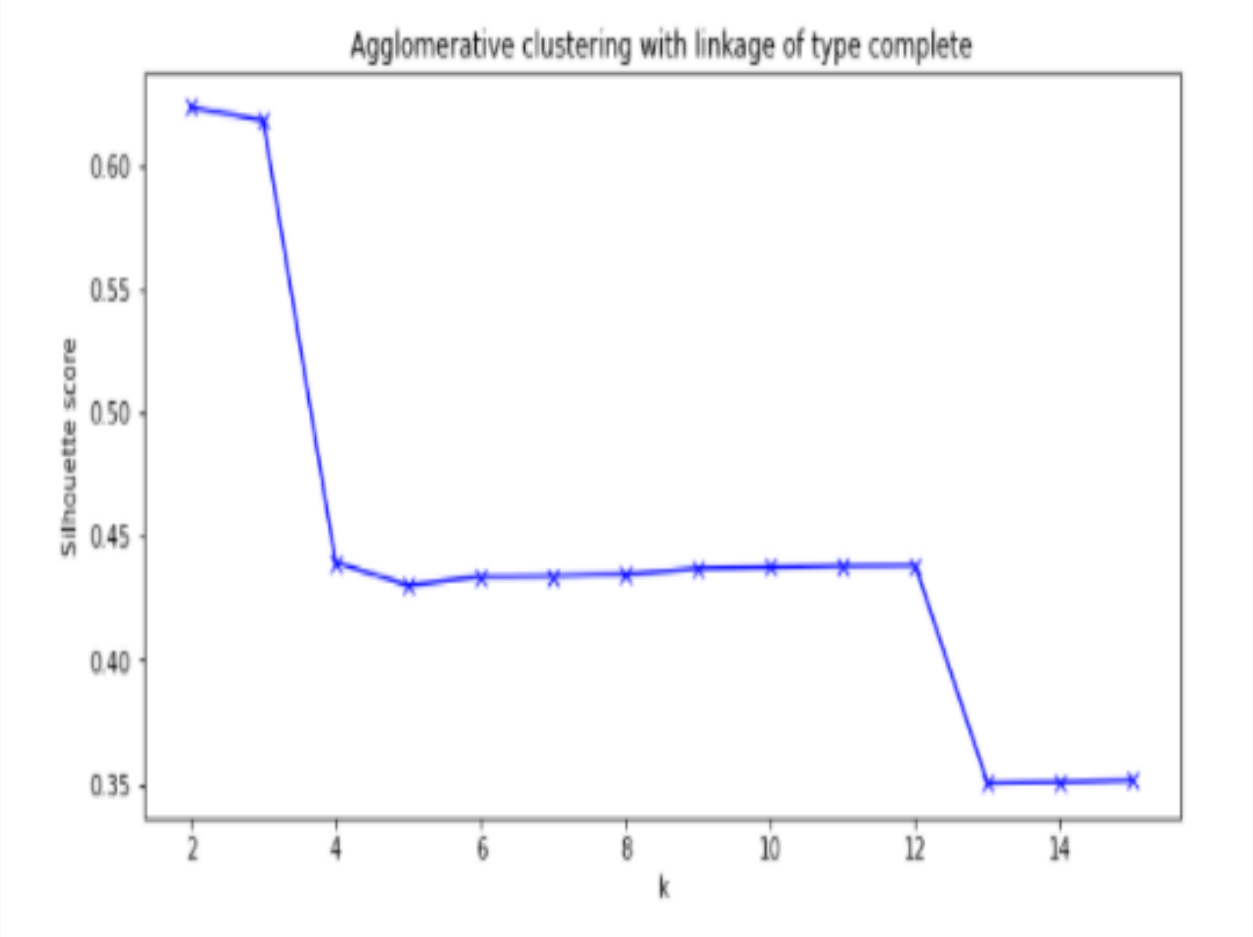
Modelowanie

Agglomerative



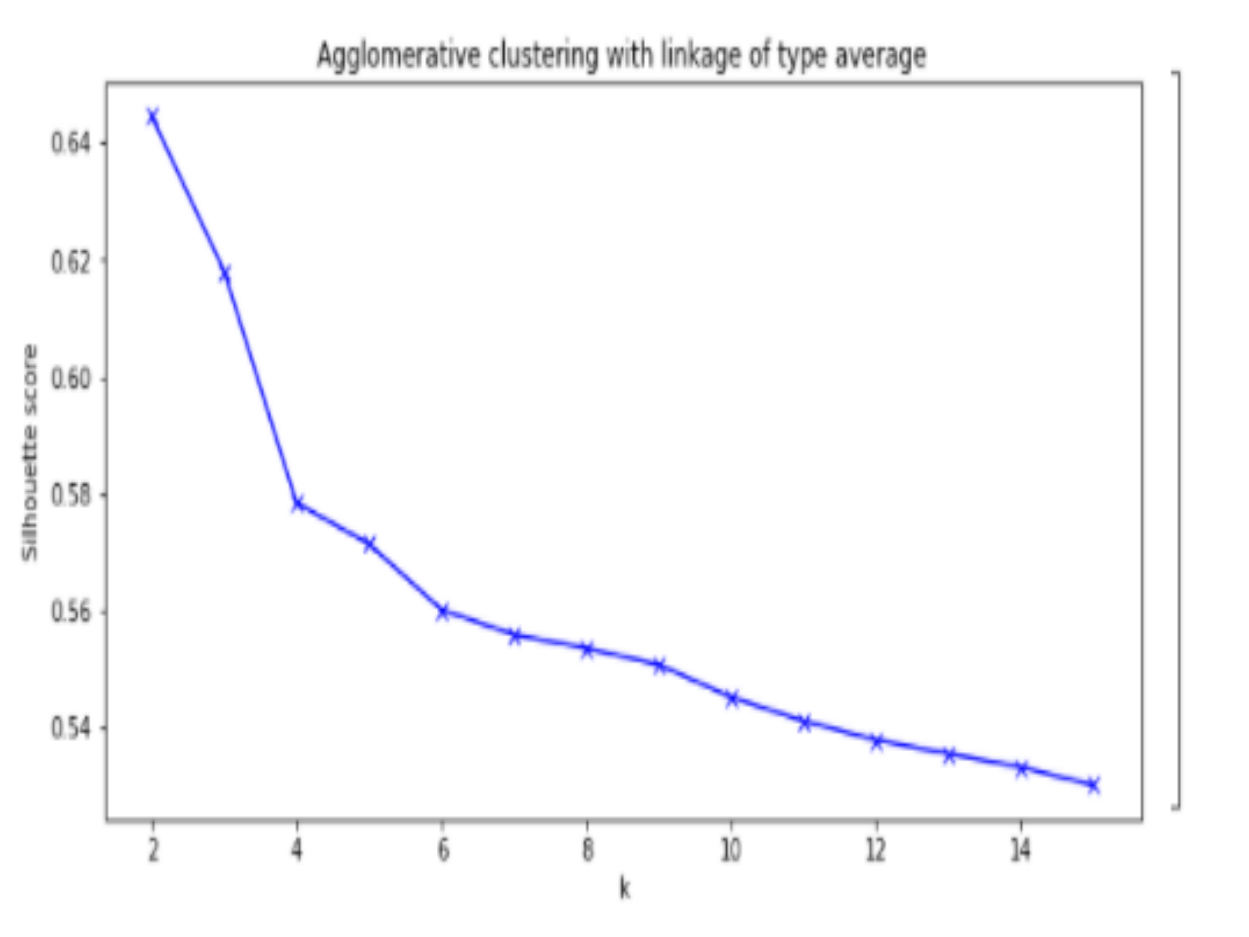
Modelowanie

Agglomerative



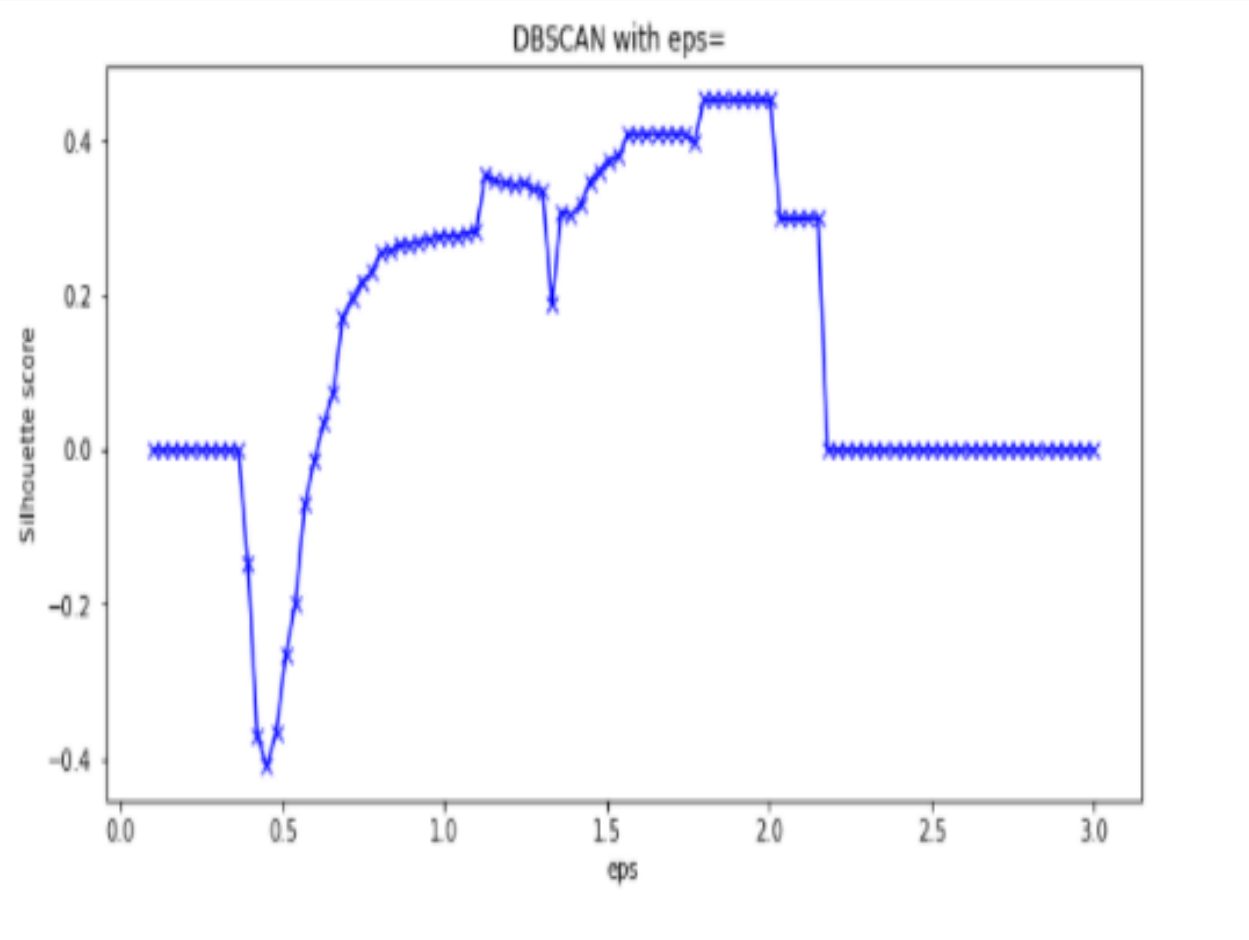
Modelowanie

Agglomerative



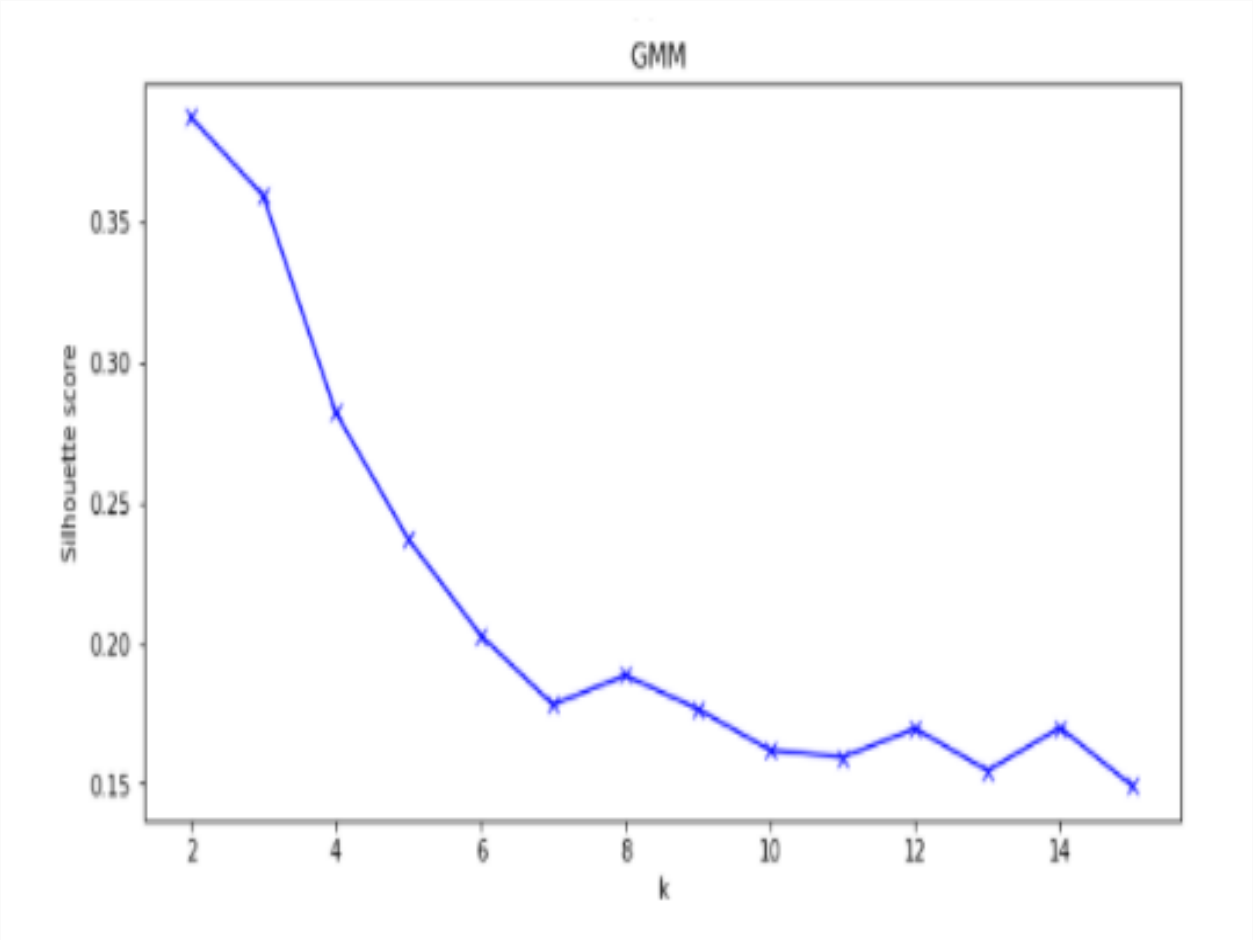
Modelowanie

DBSCAN



Modelowanie

GMM



Wyniki

Podsumowanie

Porównywaliśmy metody za pomocą wykresu łokciowego i wykresu miary Silhouette.
Silhouette score by dla własnej metryki dużo wyższy niż w przypadku metryki euklidesowej, jednak nie powinniśmy jednoznacznie stwierdzać wyższości tej metody, gdyż silhouette jest wrażliwy na metrykę użytą do jego obliczania.
Na sam koniec pojedliśmy się próby wizualizacji wyników za pomocą t-SNE i policzenia homogeniczności używając orginalnych klas.

Podsumowanie

DBSCAN wyłapał tutaj tylko jeden duży klaster i ewentualne pojedyncze klasy odstające. Nie ma sensu tutaj nawet rozważać homogenity. Nasza własna metryka pomimo dobrego silhouette daje bardzo słabe wyniki dla naszego zbioru danych. Nie potrzebujemy liczyć homogenity, żeby to zobaczyć, wystarczy spojrzeć na t-SNE.

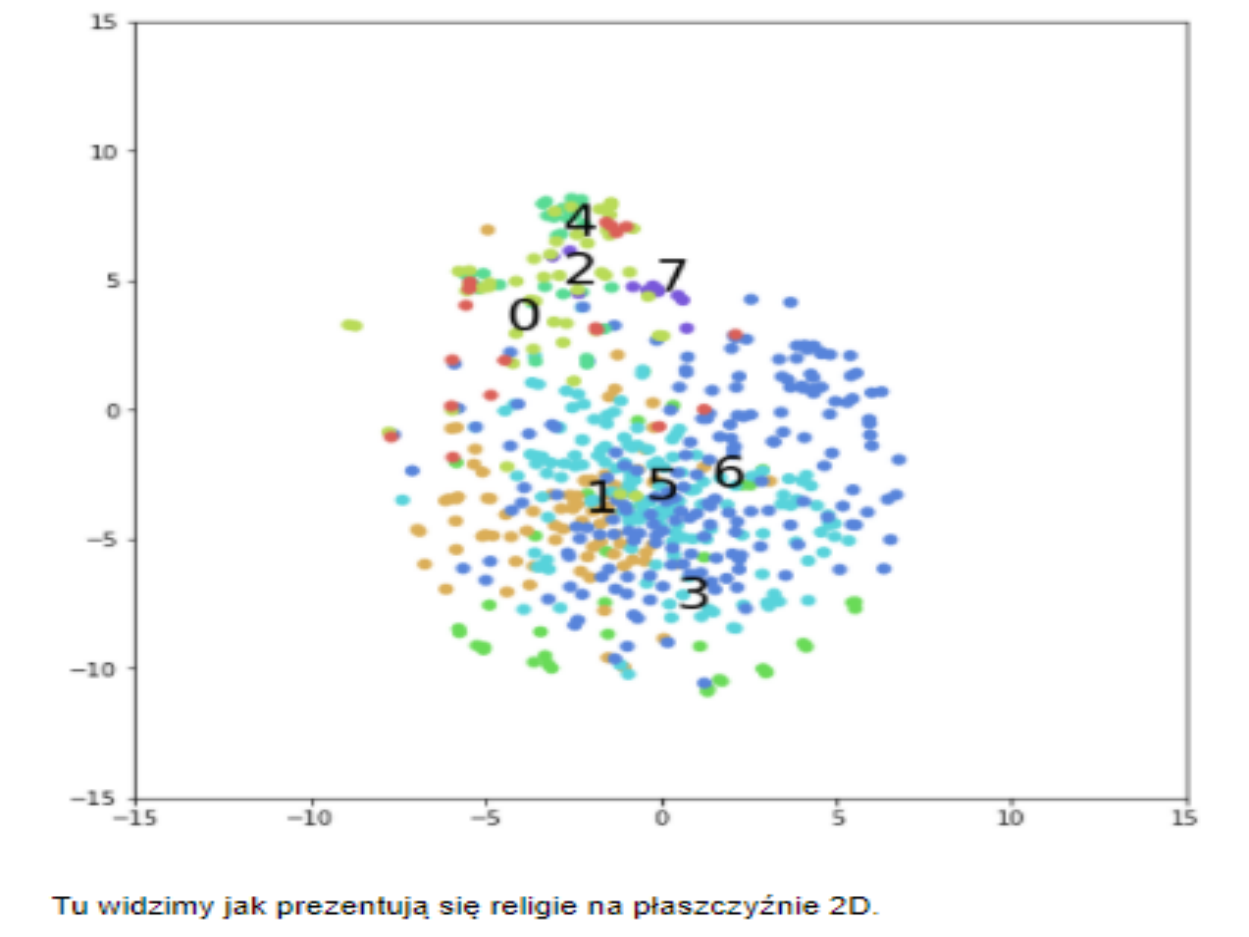
Podsumowanie

Wybraliśmy ostatecznie trzy modele do porównania:

- AgglomerativeClustering(n_clusters=6,linkage=ąverage")
  - Homogenity religion : 0.38
  - Homogenity book : 0.29
- GaussianMixture(n_components=8, covariance_type="full")
  - Homogenity religion :0.40
  - Homogenity book : 0.32
- KMeans(n_clusters=6,random_state=0)
  - Homogenity religion : 0.39
  - Homogenity book : 0.30

Podsumowanie

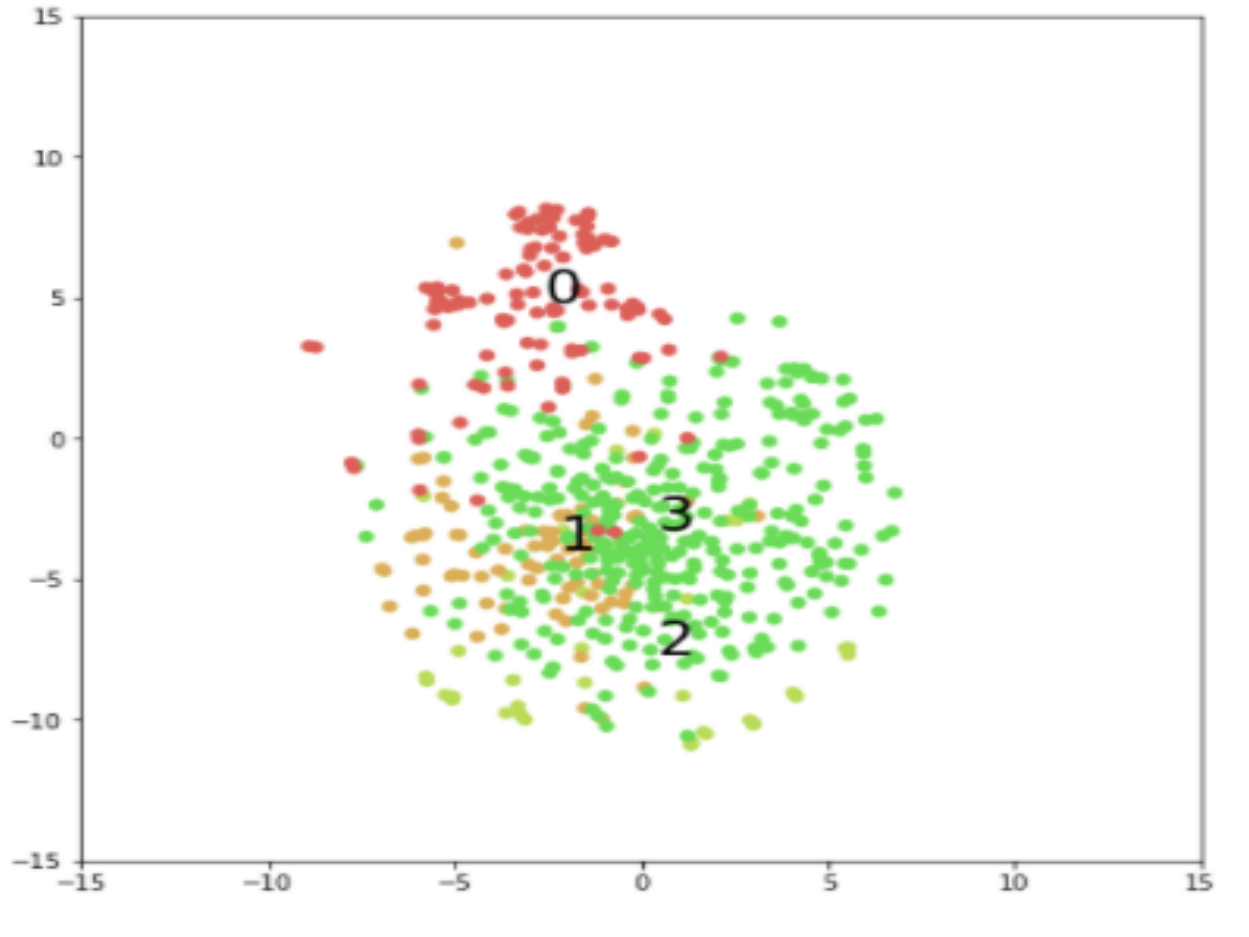
Książki na płaszczyźnie 2D



Tu widzimy jak prezentują się religie na płaszczyźnie 2D.

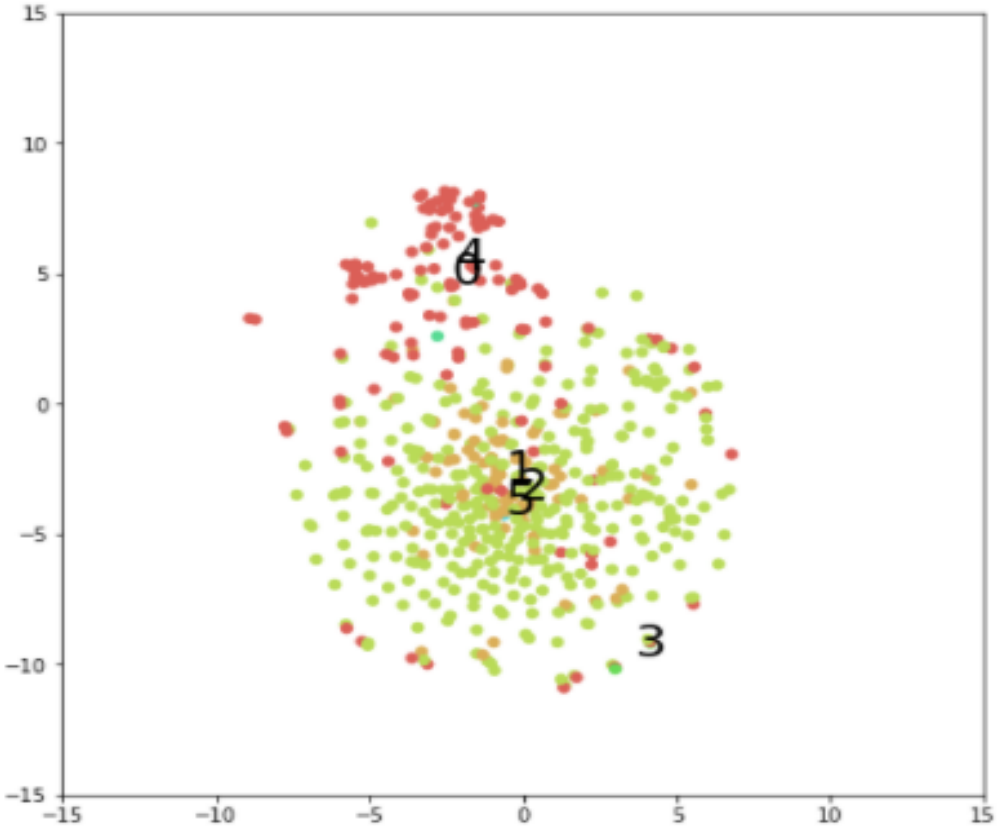
Podsumowanie

Religie na płaszczyźnie 2D



Podsumowanie

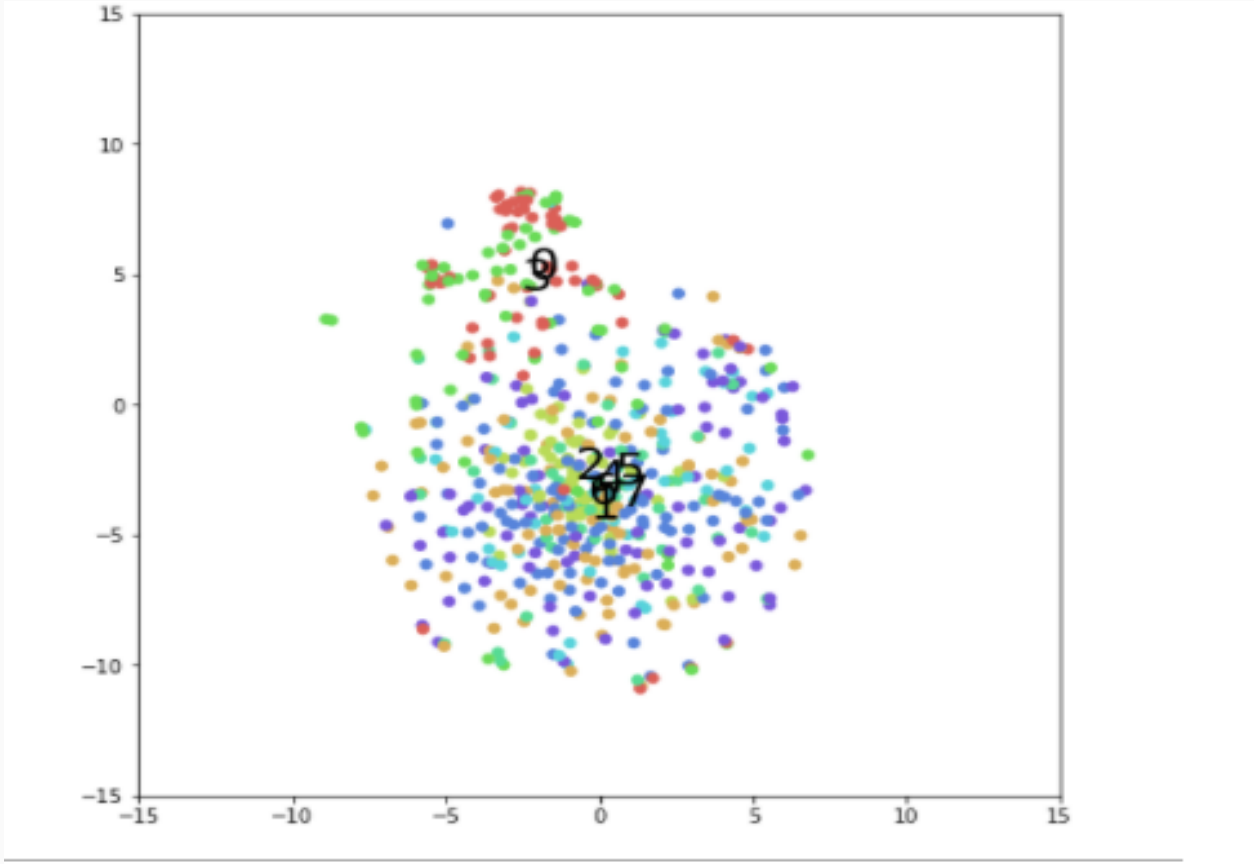
AgglomerativeClustering(n_clusters=6,linkage=ąverage")



Tak prezentuje się agglomerative clustering z metryką euklidesową, linkage=average i k=6.

Podsumowanie

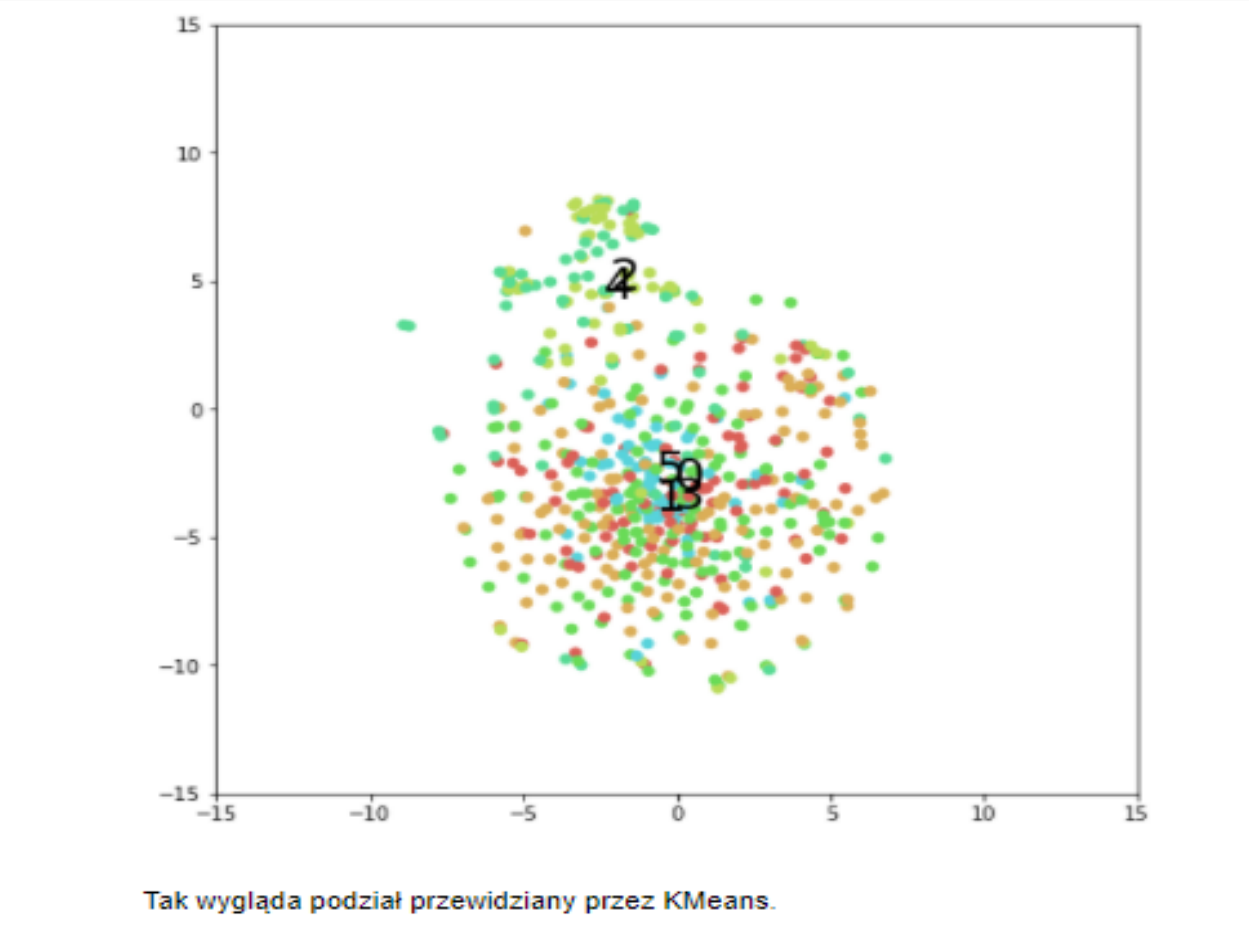
GaussianMixture(n_components=8, covariance_type="full")



Tak poradził sobie GMM z k=8,.

Podsumowanie

KMeans(n_clusters=6,random_state=0)



Tak wygląda podział przewidziany przez KMeans.

Podsumowanie

DBSCAN