

Laboratorium 4

Wprowadzenie do sztucznej inteligencji

Agnieszka Głuszkiewicz

1 Zadanie 1 - klasteryzacja k-średnich (k-means)

1.1 Wstęp

W ramach tego zadania skupiłem się na klasteryzacji zbioru danych EMNIST MNIST, wykorzystując w tym celu własną implementację algorytmu K-średnich. Aby poradzić sobie z dużym zbiorem danych, zdecydowałem się na wariant Mini-Batch K-Means, który znaczaco przyspiesza proces obliczeniowy.

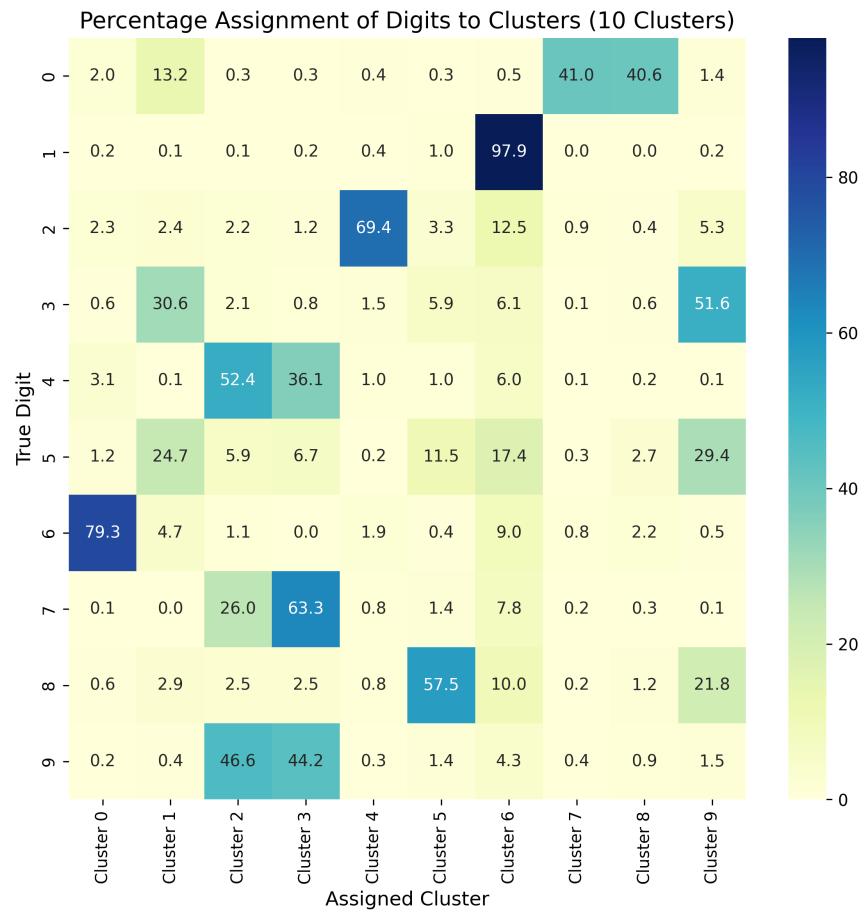
Przygotowanie danych

Do analizy użyłem pełnego zbioru danych MNIST, obejmującego 70 000 obrazów cyfr od 0 do 9. Każdy obraz, o rozmiarze 28×28 pikseli, został spłaszczony do postaci 784-wymiarowego wektora. Następnie wartości pikseli zostały znormalizowane do zakresu $[0, 1]$.

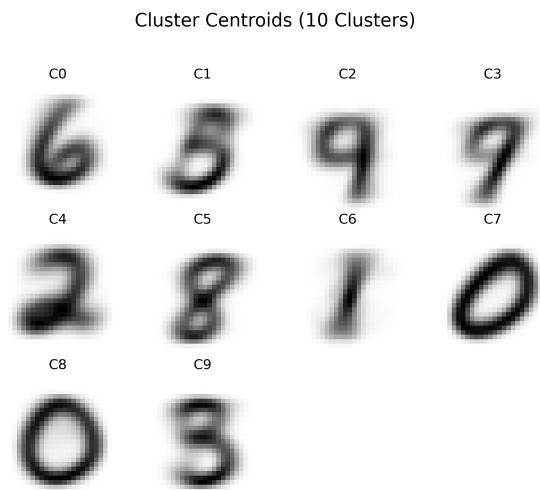
Implementacja algorytmu k-średnich (Mini-Batch K-Means)

- Zastosowałem metodę inicjalizacji centroidów **k-means++**, co pozwoliło na wybór bardziej optymalnych punktów startowych i przyspieszenie konwergencji.
- Klasteryzacja odbywa się na **mini-batchach** (losowo wybranych podzbiorach danych), co jest kluczowe dla wydajności na dużych zbiorach danych.
- Dla każdej konfiguracji parametrów algorytm jest uruchamiany **kilka razy**, a ostatecznie wybierana jest klasteryzacja o najniższej inercji.

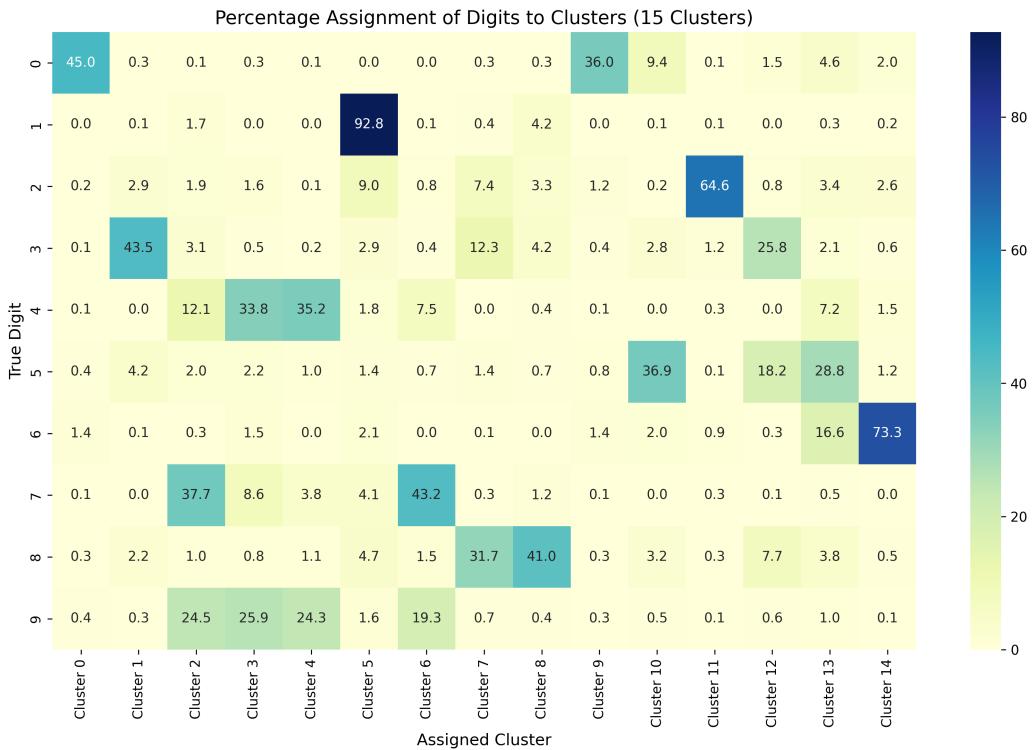
1.2 Wyniki klasteryzacji



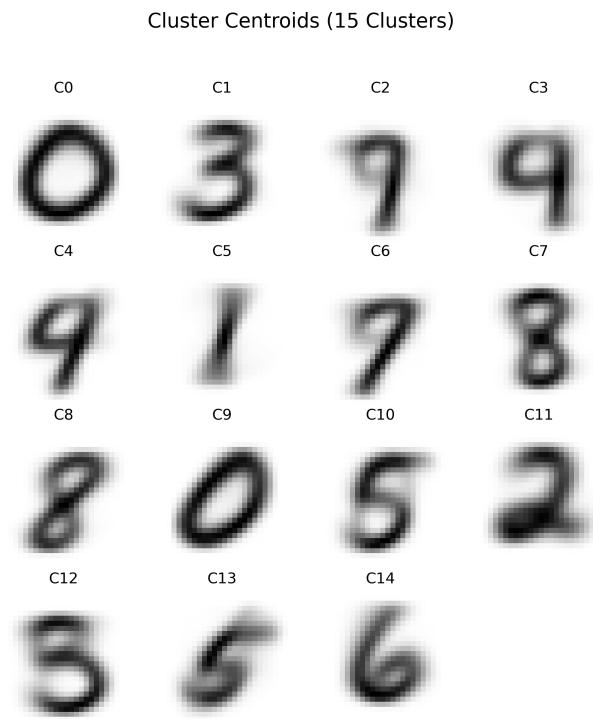
Rysunek 1: Procentowy przydział cyfr do 10 klastrów.



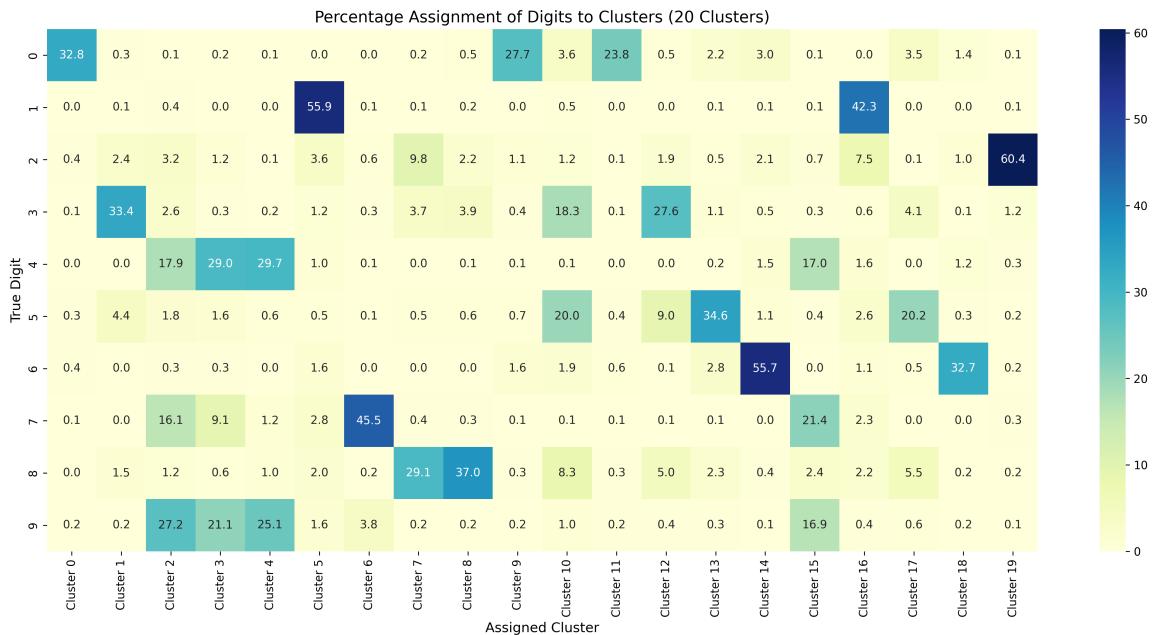
Rysunek 2: Wizualizacja centroidów dla 10 klastrów.



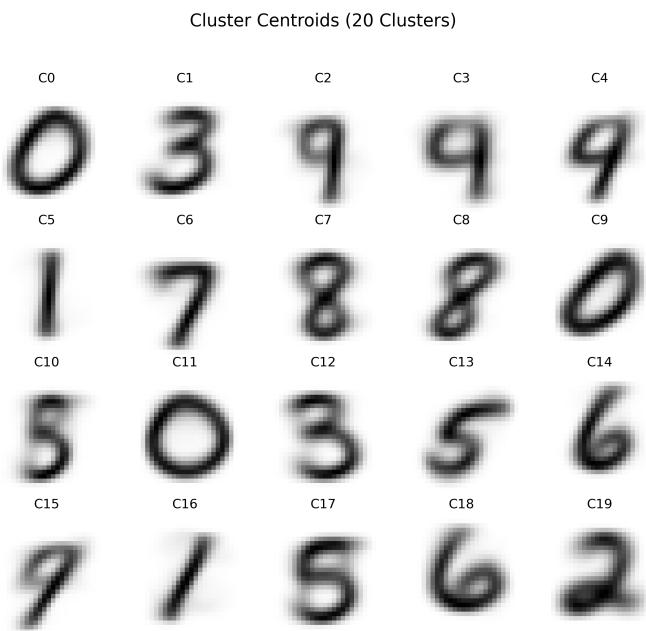
Rysunek 3: Procentowy przydział cyfr do 15 klastrów.



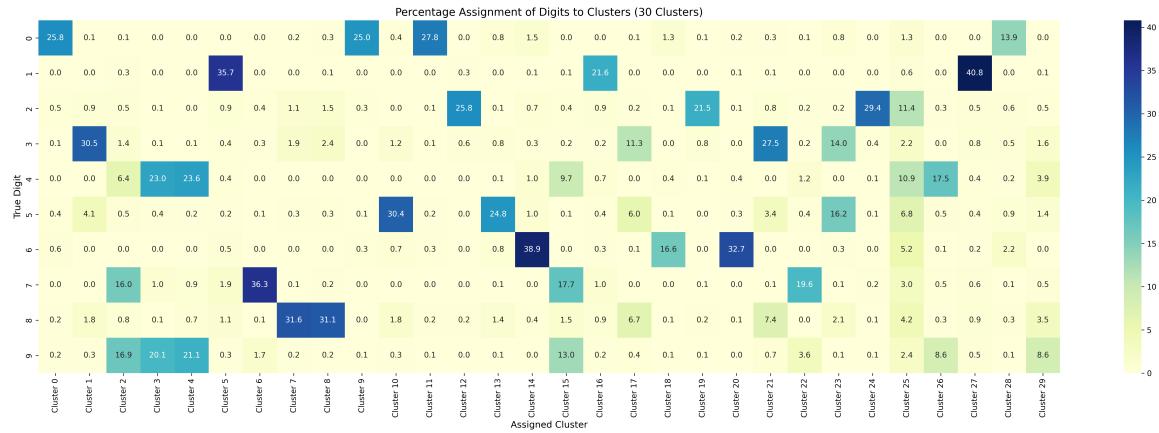
Rysunek 4: Wizualizacja centroidów dla 15 klastrów.



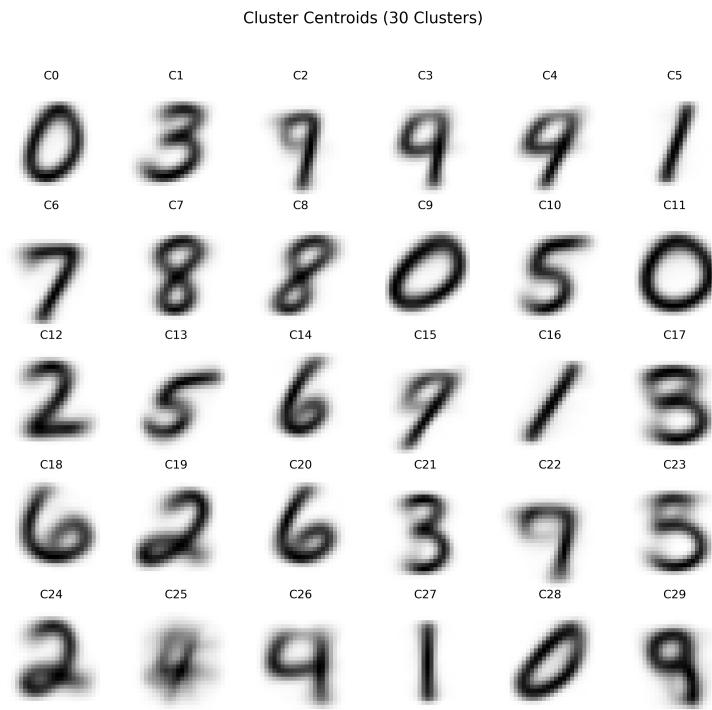
Rysunek 5: Procentowy przydział cyfr do 20 klastrów.



Rysunek 6: Wizualizacja centroidów dla 20 klastrów.



Rysunek 7: Procentowy przydział cyfr do 30 klastrów.



Rysunek 8: Wizualizacja centroidów dla 30 klastrów.

1.3 Łączenie klastrów na potrzeby klasyfikatora cyfr

Na podstawie przeprowadzonej analizy macierzy przydziału oraz wyglądu centroidów można zauważyc, że przy większej liczbie klastrów algorytm często dzieli jedną cyfrę na kilka mniejszych klastrów, zazwyczaj reprezentujących różne style pisania.

1.4 Wnioski

Podsumowując, implementacja algorytmu Mini-Batch K-Means okazała się być dość efektywna dla pełnego zbioru danych MNIST, choć nie idealna, jako że niektóre klastry zawierają w przewadze dwie lub nawet więcej cyfr, lecz wynika to z charakteru algorytmu i zapewnionych danych.

2 Zadanie 2 - klasteryzacja DBSCAN

2.1 Wstęp

Celem zadania była implementacja i ocena algorytmu klasteryzacji DBSCAN zastosowanego do zbioru danych MNIST. Moim głównym celem było uzyskanie jednolitych klastrów (zawierających w przeważającej części punkty należące do jednej klasy - cyfry), przy jednoczesnej minimalizacji liczby punktów uznanych za szum. Dążyłam również do utrzymania liczby klastrów poniżej 30 oraz otrzymania akceptowalnych wartości wskaźników oceniających jakość klasteryzacji.

2.2 Metodologia

Dane MNIST zostały załadowane ze zbioru treningowego (60.000 obrazów). Obrazy o wymiarach 28×28 pikseli spłaszczyłam do jednowymiarowych wektorów o długości 784, a następnie znormalizowałam wartości pikseli do zakresu $[0, 1]$ poprzez podzielenie przez 255.0.

Redukcja wymiarowości (PCA + t-SNE)

Aby łatwiej przetwarzać dane, zastosowałam dwuetapową redukcję wymiarowości:

1. **PCA (Principal Component Analysis):** Początkowo zredukowałam wymiarowość danych do 50 komponentów. Ten krok, będący liniową transformacją, pomógł w usunięciu redundancji między cechami i wstępny odszumianiu danych.
2. **t-SNE (t-distributed Stochastic Neighbor Embedding):** Następnie, wynik PCA (50-wymiarowy) został dalej zredukowany do 2 wymiarów. t-SNE to nielinowa technika, która służy do wizualizacji danych o wielu wymiarach poprzez odwzorowanie struktur sąsiedztwa w przestrzeni o mniejszej liczbie wymiarów.

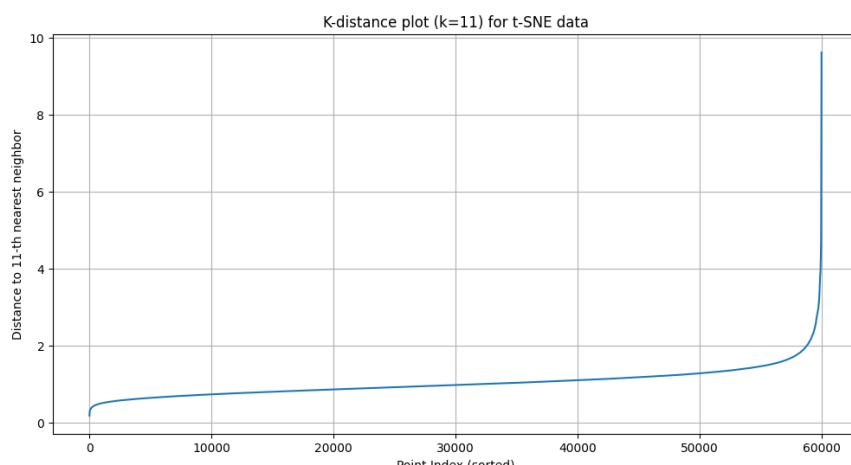
Ostateczna 2-wymiarowa reprezentacja danych uzyskana za pomocą t-SNE posłużyła jako wejście dla algorytmu DBSCAN. Wygenerowałam również wykres punktowy przedstawiający dane po t-SNE, pokolorowane według ich prawdziwych etykiet, co pozwoliło na wstępna wizualną ocenę separacji cyfr.

Implementacja DBSCAN i dobór parametrów

Wykorzystałam moją własną implementację algorytmu DBSCAN. Kluczowe parametry tego algorytmu to:

- `eps = 2.0`: maksymalna odległość między dwoma próbками, aby jedna mogła być uznana za sąsiada drugiej.
- `min_samples = 11`: minimalna liczba próbek w sąsiedztwie punktu, aby ten punkt został uznany za rdzeniowy.

W doborze optymalnych parametrów kierowałam się metodą prób i błędów oraz analizą wykresu K-distance, który dla danych po t-SNE, poprzez wizualne zidentyfikowanie "łokcia" (ang. "elbow"), sugeruje odpowiednią wartość `eps`.



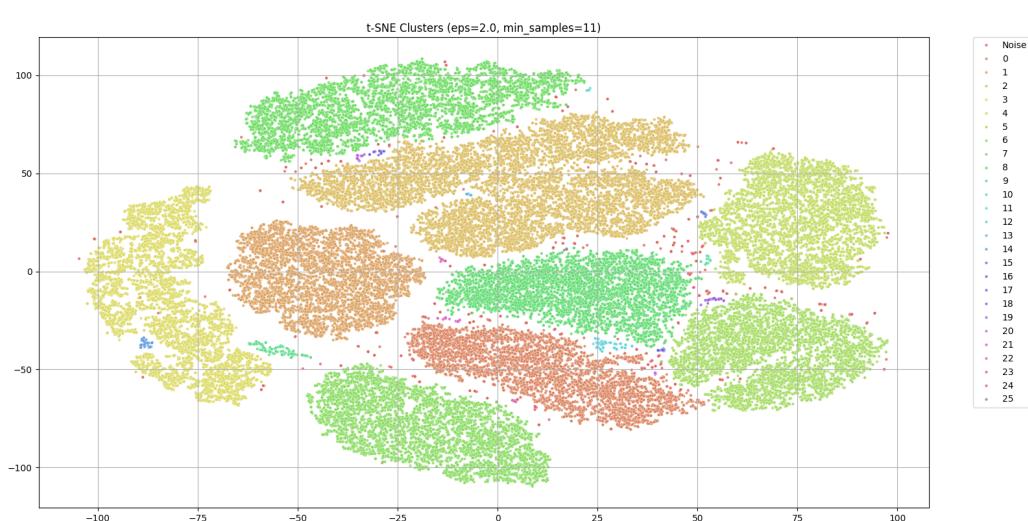
2.3 Wyniki

Dzięki zastosowanym przeze mnie parametrom (`eps = 2.0` i `min_samples = 11`) na 60 000 próbkach MNIST, po redukcji liczby wymiarów otrzymałem następujące wyniki:

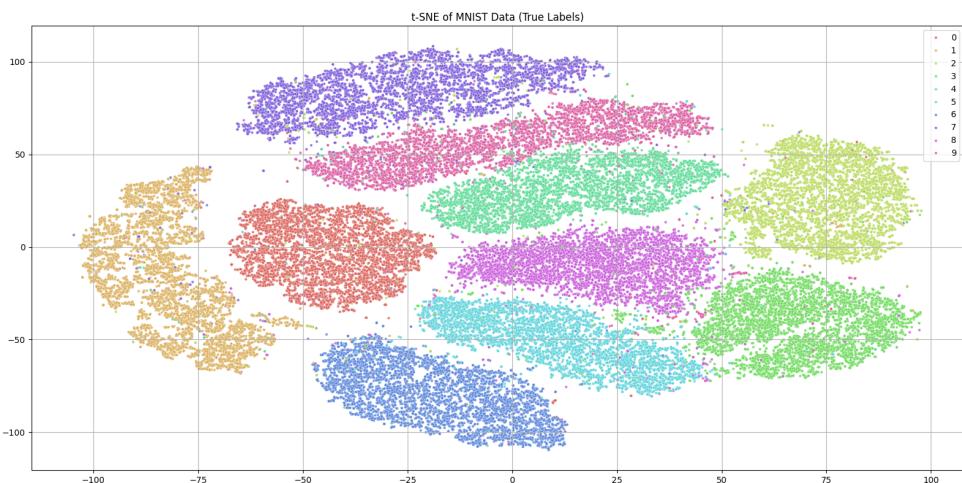
- Liczba znalezionych klastrów: **26**
- Szum: **0.0081**
- Homogeniczność: **0.8845**
- Kompletność: **0.9193**
- V-measure: **0.9015**
- Adjusted Rand Index (ARI): **0.8525**
- Fowlkes-Mallows Score: **0.8713**

Wysoka wartość homogeniczności (**0.8845**) wskazuje, że wyznaczone klastry w przeważającej większości zawierają punkty należące do jednej prawdziwej cyfry. To był jeden z kluczowych celów.

Liczba uzyskanych klastrów (**26**) mieści się w założonym zakresie (poniżej 30). Niski procent szumu (**0.81%**) świadczy o udanym przypisaniu większości punktów do klastrów.



Rysunek 9: Wykres przedstawiający klastry DBSCAN (eps=2.0, min_samples=11).



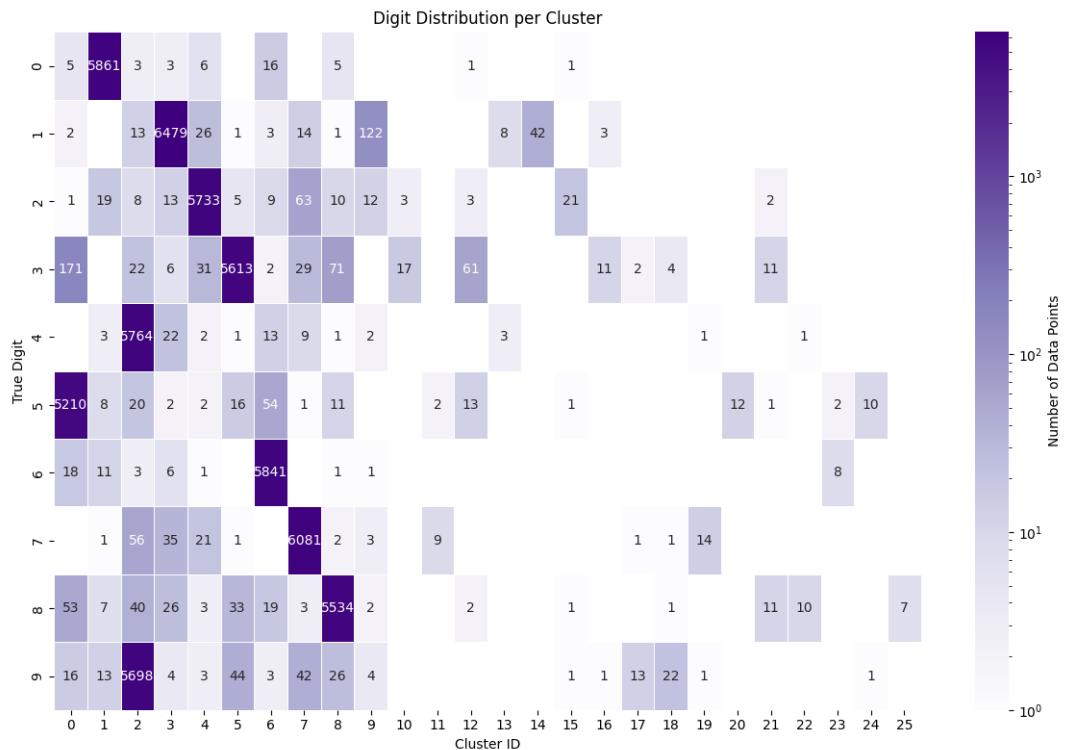
Rysunek 10: Wykres przedstawiający prawdziwe klastry przyporządkowane do cyfr, dla porównania.

Ocena dokładności klasyfikacji i błędne przyporządkowania

Aby ocenić, jak dobrze klastry DBSCAN odpowiadają rzeczywistym cyfrom, każdemu klastrowi przypisuję cyfrę, która była w nim najliczniejsza. Następnie obliczyłem dokładność klasyfikacji jako stosunek liczby poprawnie przypisanych punktów do ogólnej liczby punktów sklastrowanych (bez szumu).

- Liczba poprawnie sklastrowanych punktów: **52514**
- Całkowita liczba sklastrowanych punktów: **59514**
- Dokładność klasyfikacji: **88.24%**
- Procent błędnych klasyfikacji w wyznaczonych klastrach: **11.76%**

Procent błędnych klasyfikacji (**11.76%**) oznacza, że tyle procent punktów, które zostały przypisane do klastrów, miało rzeczywistą etykietę inną niż dominująca etykieta krastra. Może to wynikać z nakładania się cyfr w przestrzeni t-SNE lub z faktu, że niektóre cyfry mają podobny kształt (np. 4 i 9).



Rysunek 11: Heatmapa rozkładu prawdziwych cyfr w poszczególnych klastrach.

2.4 Wnioski

Moja implementacja algorytmu DBSCAN, poprzedzona redukcją liczb wymiarów z wykorzystaniem PCA i t-SNE, okazała się całkiem skuteczna w klasteryzacji zbioru danych MNIST. Wybrane parametry, `eps=2.0` i `min_samples=11`, pozwoliły mi uzyskać klastry o wysokiej homogeniczności (**0.8845**), z bardzo niskim poziomem szumu (**0.0081**), rozsądną liczbą klastrów (**26**) oraz dobrą dokładnością klasyfikacji dla punktów sklastrowanych (**88.24%**).