AGH

AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA
W KRAKOWIE

# Metody analizy i wizualizacji dużych zbiorów danych

## Laboratorium 5 - Gephi

**Jan Gołda, Wojciech Basiura - WIET 2018/2019**

# 1. Opis zagadnienia

Laboratorium 5 poświęcone było wizualizacji grafów w programie Gephi.

Zadaniem domowym była wizualizacja zbioru Fashion-MNIST jako grafu KNN oraz analiza otrzymanych wyników.

## 1.1. Zbiór danych

Wizualizowanym zbiorem jest Fashion-MNIST dostępny pod adresem:

https://www.kaggle.com/zalando-research/fashionmnist

Jest to zbiór czarno-białych obrazków przedstawiających elementy garderoby, które podzielone są na następujące klasy:

0. T-Shirt / top
1. Trouser
2. Pullover
3. Dress
4. Coat
5. Sandal
6. Shirt
7. Sneaker
8. Bag
9. Ankle boot

Pobrany zbiór składa się z 10000 obrazków, co znacznie przerasta możliwości programu Gephi (problemy wynikające z dostępną pamięcią), dlatego też zdecydowaliśmy się ograniczyć dane do 4000 losowo wybranych obrazków.

# 2. Generacja grafu

Aby móc zwizualizować zbiór jako graf przekształciliśmy go w graf 40 najbliższych sąsiadów, gdzie jako odległość między sąsiadami przyjęliśmy odległość euklidesową pomiędzy wektorami będącymi reprezentacją nasycenia poszczególnych pikseli obrazka.

Generacji grafu dokonaliśmy w języku Python z wykorzystaniem funkcji NearestNeighbours z biblioteki scikit-learn. Pełny kod generujący graf dostępny jest w repozytorium.

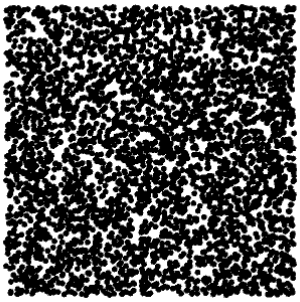
Utworzony graf miał następującą postać:

4000 węzłów, każdy posiada id oraz numer i nazwę klas do której należy

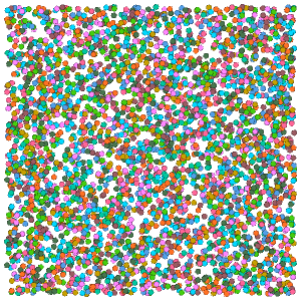
16000 krawędzi, 40 krawędzi skierowanych dla każdego węzła

## 3. Wizualizacja

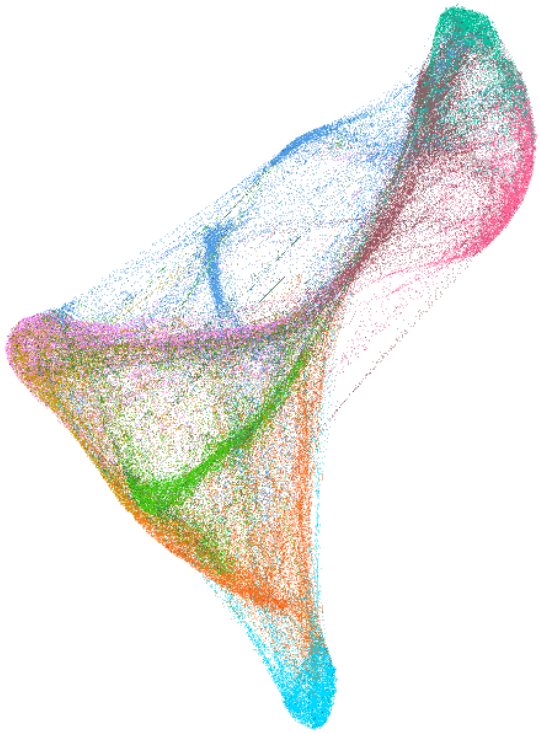
Po zaimportowaniu graf przedstawiał się następująco:



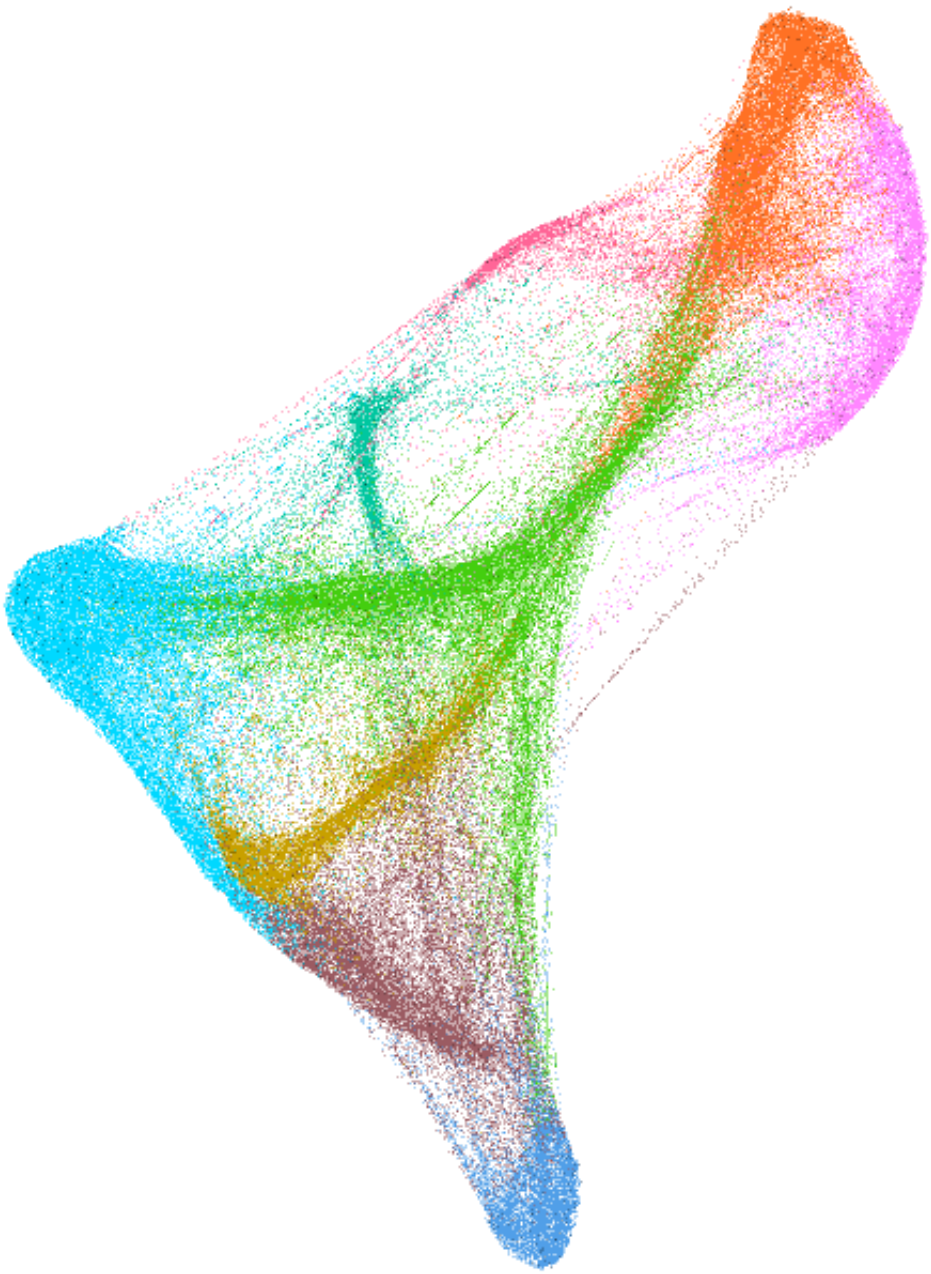
Po pokolorowaniu wierzchołków na podstawie klas do których należą:



W celu uzyskania lepszego układu wierzchołków na płaszczyźnie użyliśmy funkcji **ForceAtlas2** (z domyślnymi parametrami), czego efekt wyglądał następująco:



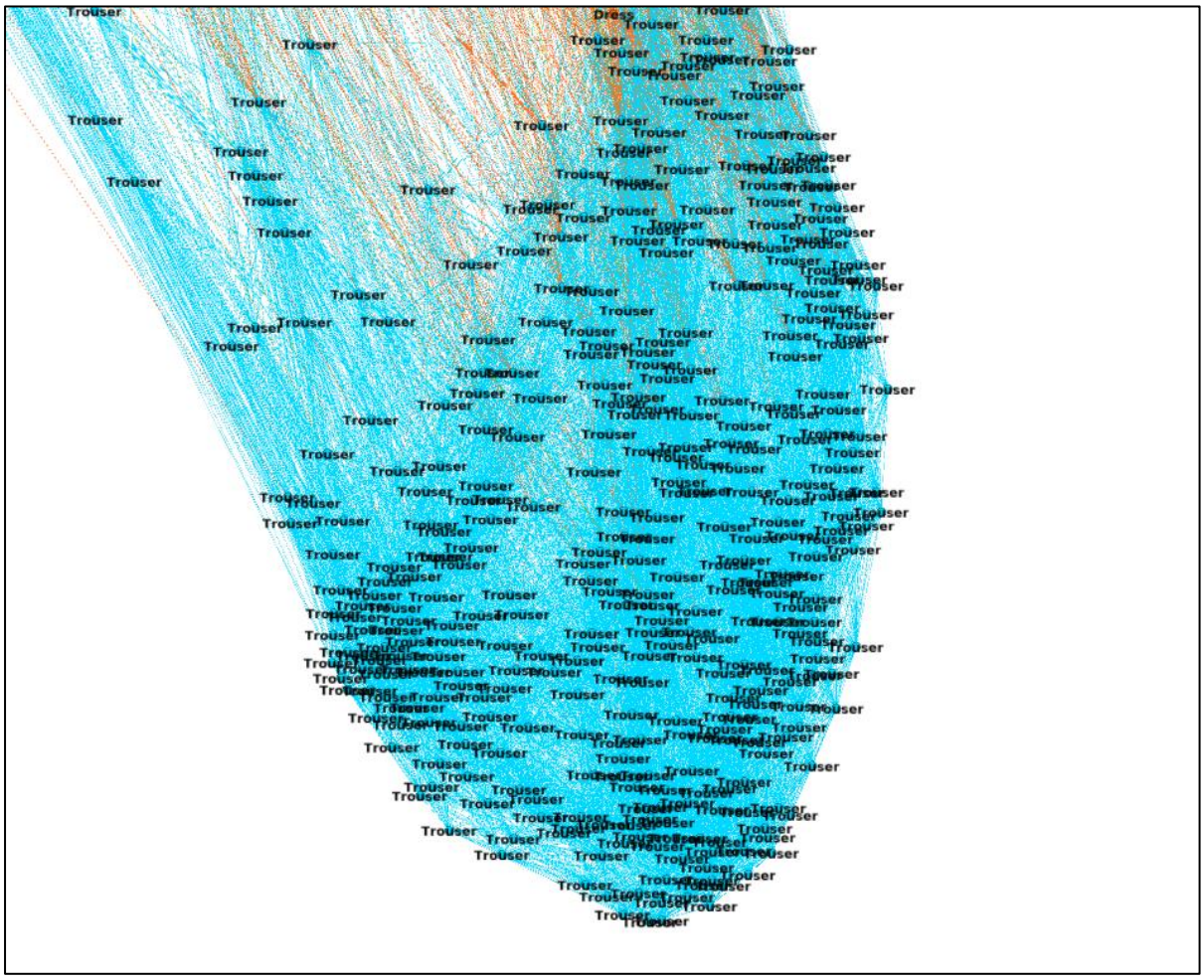
Jako że na powyższym grafie widać, że klasy się mieszają, postanowiliśmy wygenerować drugie kolorowanie przy użyciu funkcji modularity (z parametrem resulution równym 1, co powinno zminimalizować liczbę wydzielonych klastrów), która dzieli wierzchołki wg ich aktualnej pozycji, a nie klasy. Wygenerowane i pokolorowane klastry wyglądają następująco:



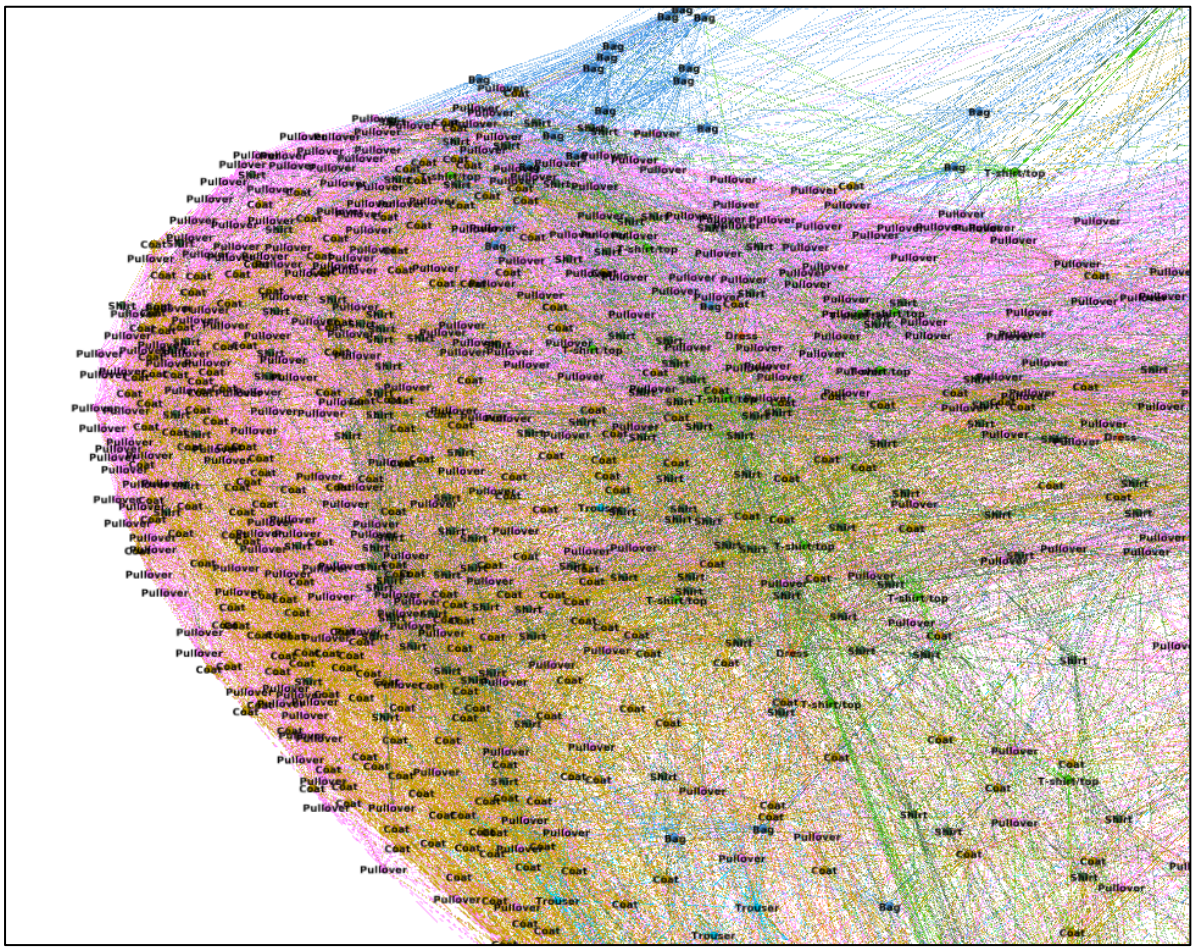
## 4. Analiza wyników

Porównanie dwóch otrzymanych grafów pozwala zauważyć, że klastry utworzone na podstawie pozycji generalnie pokrywają się z klasami ze zbioru danych, lecz w niektórych miejscach klasy się mieszają.

Przykładem klasy, która praktycznie pokrywa się z uzyskanym klastrem jest klasa spodni. Co nie dziwi zważywszy że zdecydowanie wyróżniają się one kształtem na tle innych klas.



Z kolei jednymi z najbardziej pomieszanych klas są wszelkie okrycia wierzchnie, w tym płaszcze, koszule i swetry, co również nie dziwi, jako że są podobne kształtem i proste KNN nie potrafi ich odróżnić.




## 5. Wnioski

Program Gephi jest przydatnym narzędziem do wizualizacji grafów z uwagi na jego dosyć rozbudowany interface oraz ilość dostępnych funkcji. Mając na uwadze że proste porównywanie odległości pomiędzy wektorami reprezentującymi nasycenia obrazów nie jest optymalnym sposobem porównywania zdjęć, można uznać że program dobrze sobie poradził z postawionym przed nim zadaniem.

Jedynym znaczącym problemem na który się natknęliśmy były ograniczenia pamięciowe Gephi, nie pozwalające na reprezentację całego zbioru. Nawet po ograniczeniu zbioru do 4000 węzłów i 40 najbliższych sąsiadów program kilkukrotnie sygnalizował przekroczenie pamięci i konieczność restartu, co było bardzo irytujące.