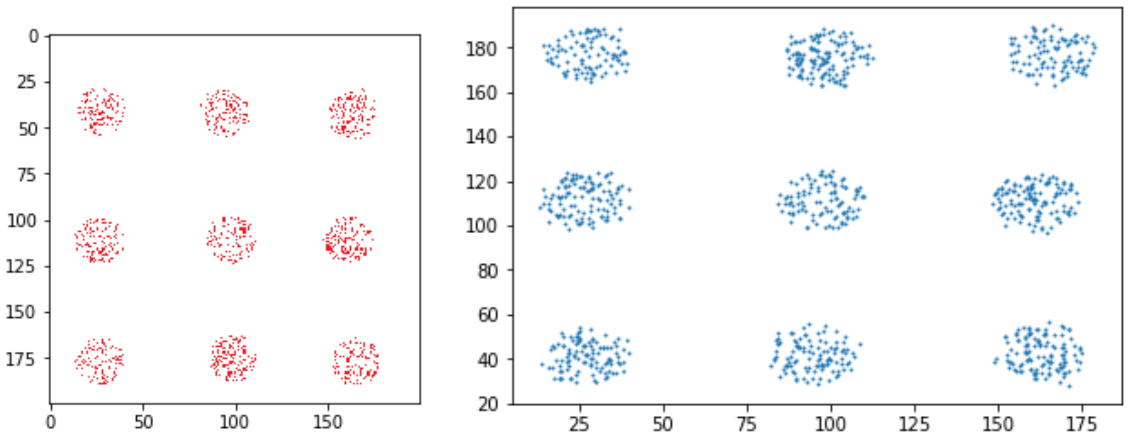
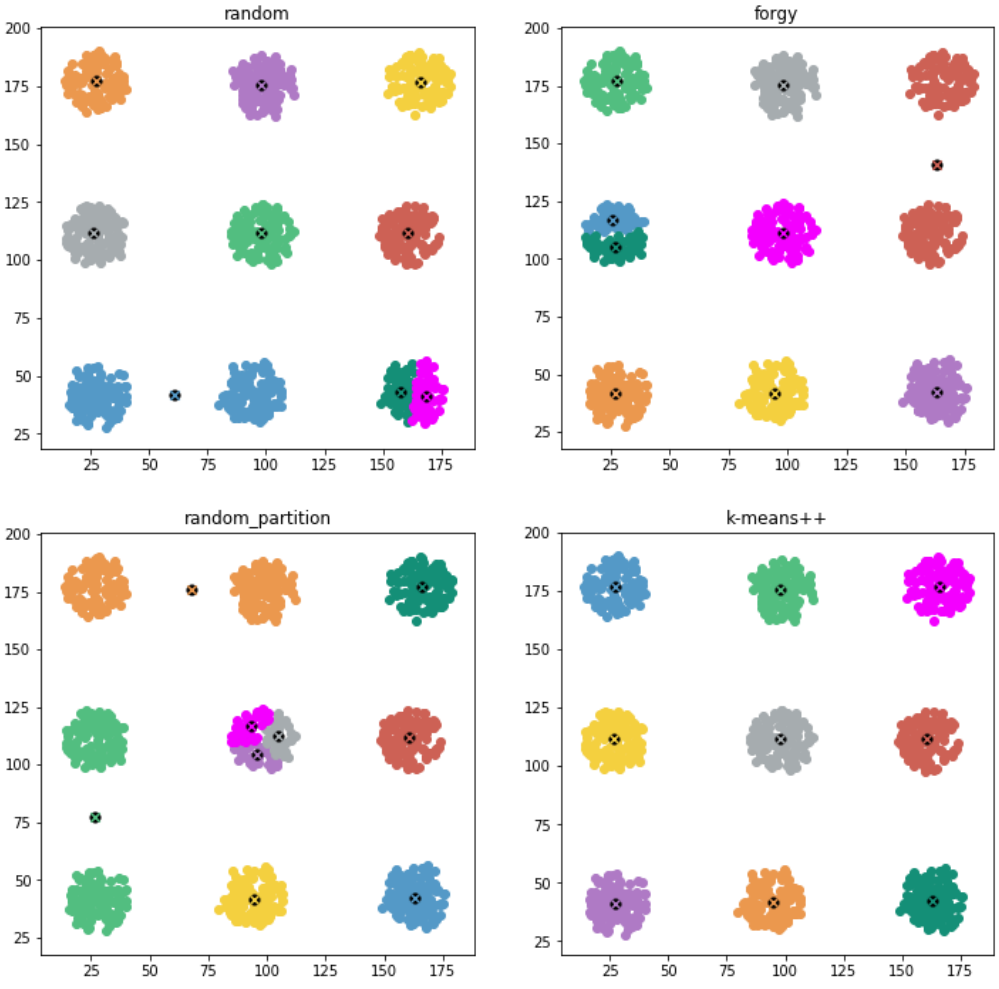
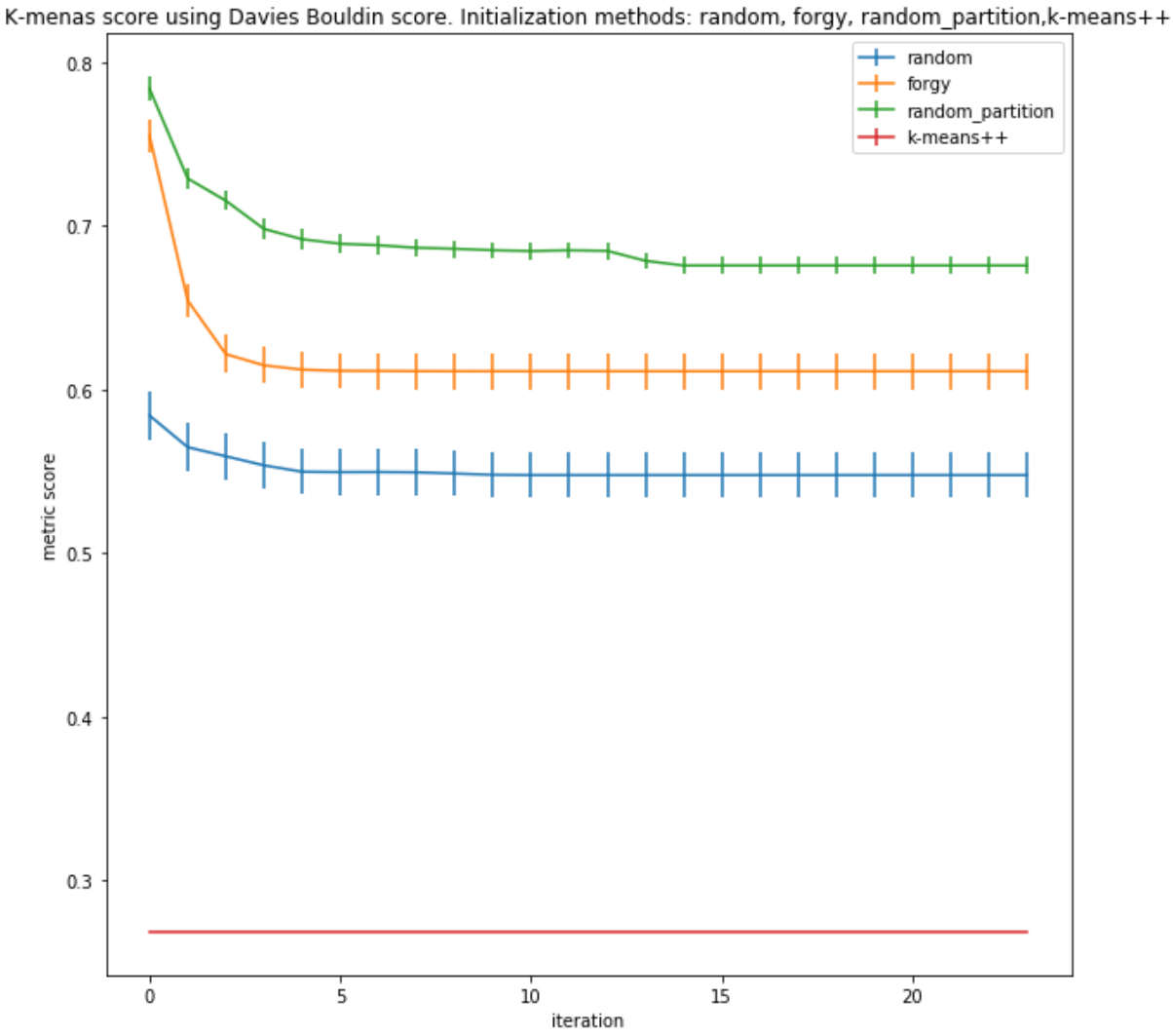
Raport – Kmeans initializations
Podstawy nauczania maszynowego
Wyk. Mateusz Woś

Dane do zadania ponownie wygenerowałem używając Painta. Zapisałem je do pliku aby w przyszłości łatwo z nich ponownie skorzystać.



Korzystając z gotowej implementacji k-means z scikit-learn zwizualizowałem klasteryzacje dla każdej z metod inicjalizacji środków klastrów.



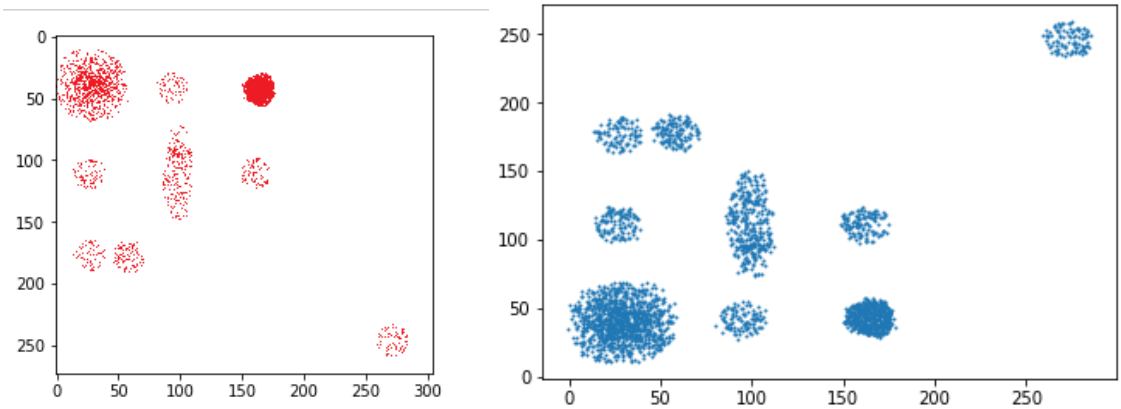


Korzystalem z metryki Daviesa Boulidna (mniejszy score -> lepiej)

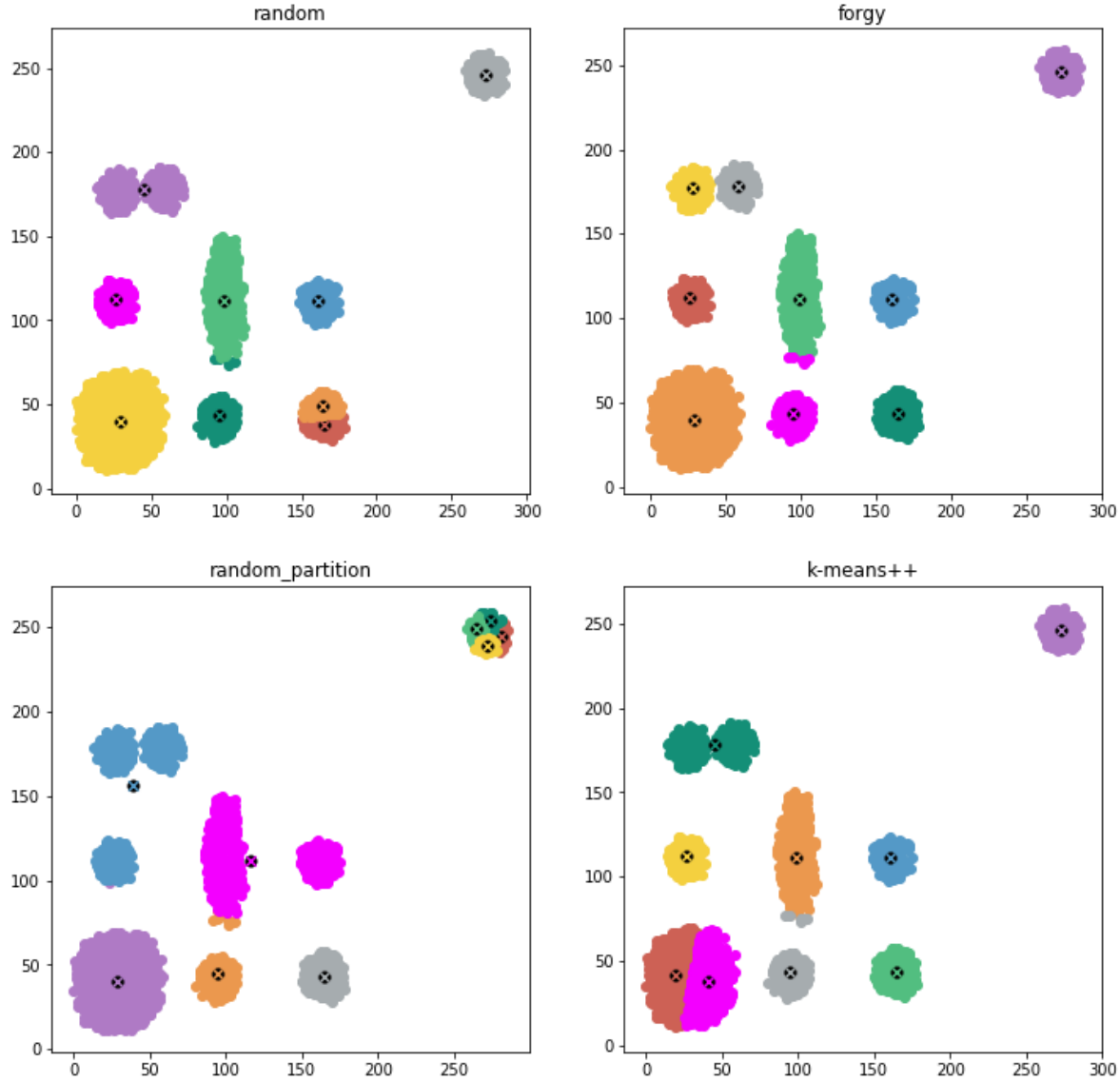
Random partition wypadł w tym przypadku najgorzej, lecz jak to bywa z losowością, czasami ta metoda bywała równie dobra jak czy kmeans++.

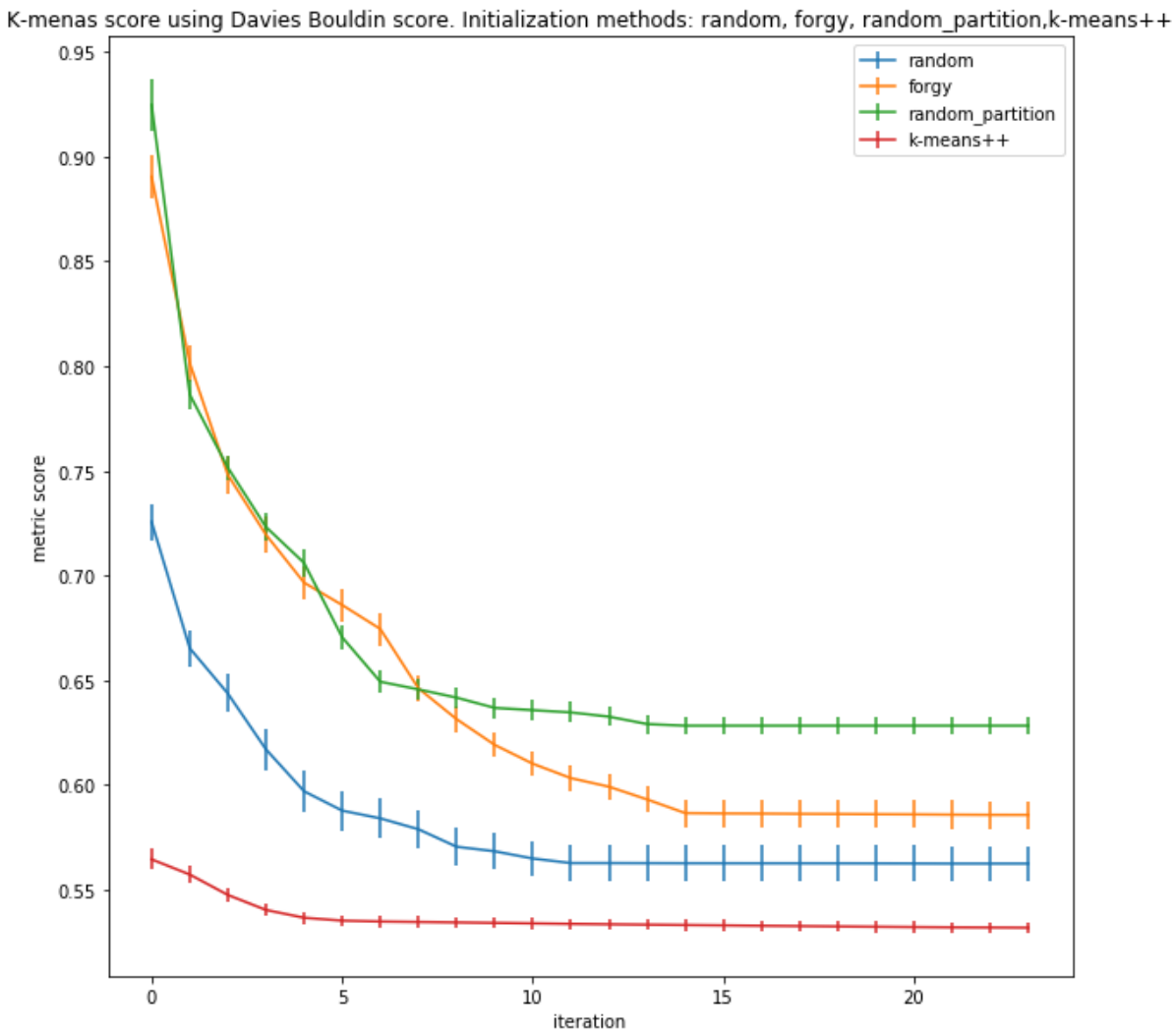
Kmeans++ jak widać powyżej jest bezkonkurencyjny. Praktycznie dla każdej ilości iteracji osiagal swój najlepszDla zadanego datasetu przy około 10 próbach zawsze idealnie wykonywał klasteryzacje.

Wykresy dość szybko osiągają swój najlepszy wynik, powyżej około 5 iteracji każda z metod już nie stawała się lepsza. Dataset z którego korzystaliśmy był dość łatwy do klasteryzacji.



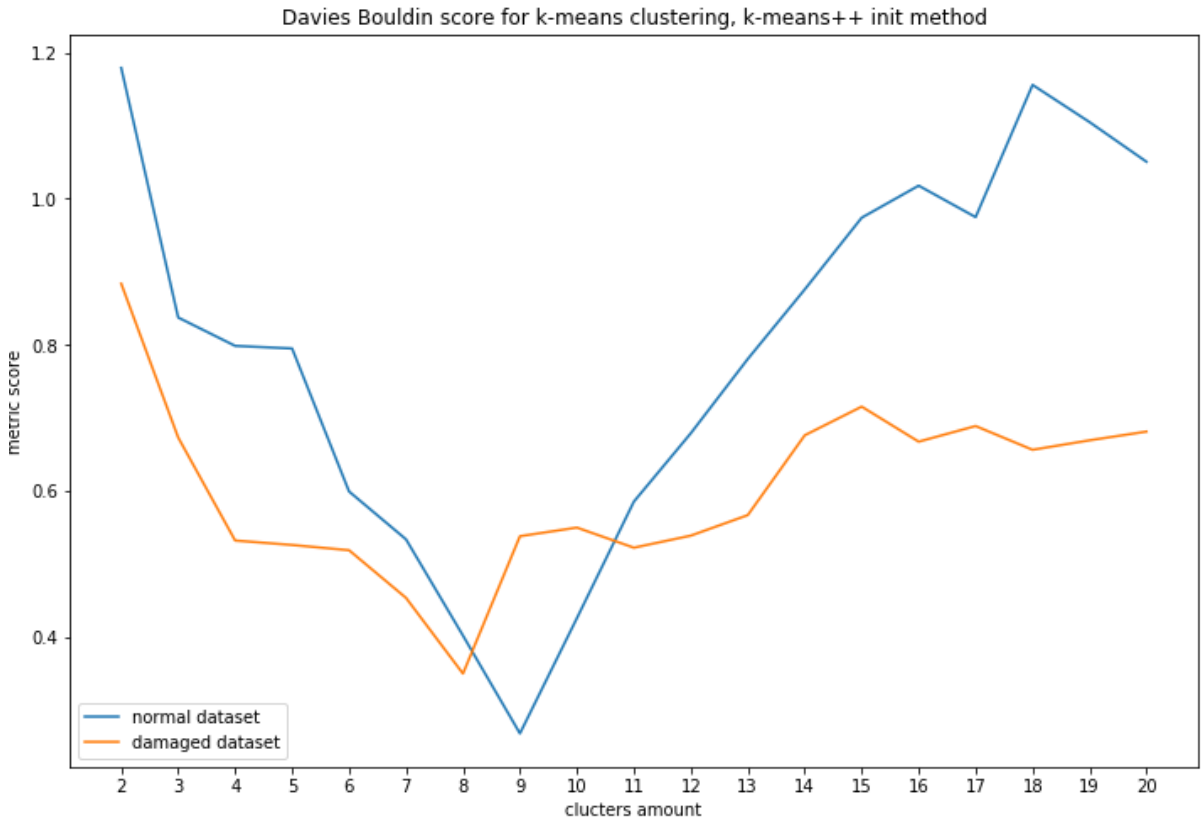
K-menas clasterization depending on initilization method





Dla zniekształconego datasetu sytuacja jest już ciekawsza. Widać tutaj, ze metody nie spłaszczają się od razu. Metody random i random partition dopiero uzyskują swoje najlepsze możliwe wyniki przy około 20 iteracja na uruchomienie. Jak i poprzednio kmeans++ jest bezkonkurencyjny.

Dodatkowo ciekawe zależności można zaobserwować na wykresie z klastrami. Dobrze można zauważyć tutaj słabości algorytmu kmeans np. zbyt duży obszar i źle wylosowane centroidy -> kmeans uznaje taki klaster jako 2 oddzielne klastry; źle wylosowane punkty(patrz: random partition) i daleko oddalony kluster -> reszta klustrow nie brana pod uwagę bo do nich jest za daleko od początkowych centorid



„Czy na ich podstawie można stwierdzić, że optymalne k to 9 (bo tyle mamy klastrów)?”
Tak, na powyższym wykresie widać, ze najlepszy wynik dla normalnego dataset (w sumie dla uszkodzonego prawie tez) jest dla k=9. Osobiście spodziewałem się większego, optymalnego k dla datasetu uszkodzonego.

Wnioski:

- Algorytm kmeans jest bardzo prosty i mało złożony, dzięki czemu nadaj się idealnie do klasteryzacji dużych datasetow.
- Niestety nie radzi sobie za bardzo z datasetami, w których nie ma wyraźnie wyróżnionych klastrów lub występują liczne anomalie.
- Musimy wiedzieć z góry ilu klastrów spodziewamy się w danym datasecie.
- Kmeans w zależności od metody inicjalizacji początkowych punktów, może dawać dość nieoczekiwane, niepoprawne wyniki. (dobrze to widać w metodzie random partition)