

Miniprojekt 3

Julia Biały

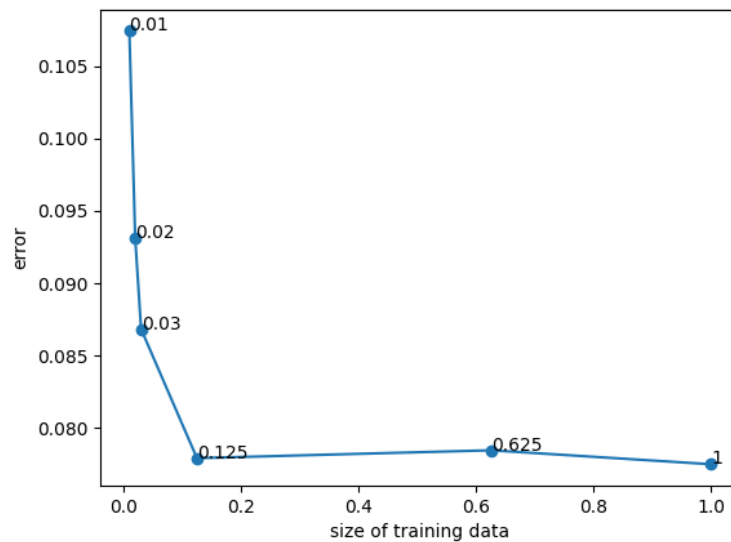
23 maja 2022

1 Model SVM

Zbiór danych podzieliłam wybierając do zbioru treningowego losowo 0.7 danych, resztę do zbioru testowego. Podczas trenowania zbioru użyłam metody stochastycznego gradientu. Wyniki zostały uśrednione dla 5 przebiegów algorytmu.

Błędy klasyfikacyjne dla kolejnych frakcji zbioru treningowego wynoszą: 0.1074, 0.0930, 0.0868, 0.0779, 0.0784, 0.0775.

Learning rate ustawiłam na 0.001, a lambdę na 0.01. Testowałam różne parametry ręcznie i przy takich były najlepsze wyniki. Zastosowałam liniową funkcję jądrową. Poniższy wykres przedstawia krzywą błędu na partycjach zbioru treningowego.

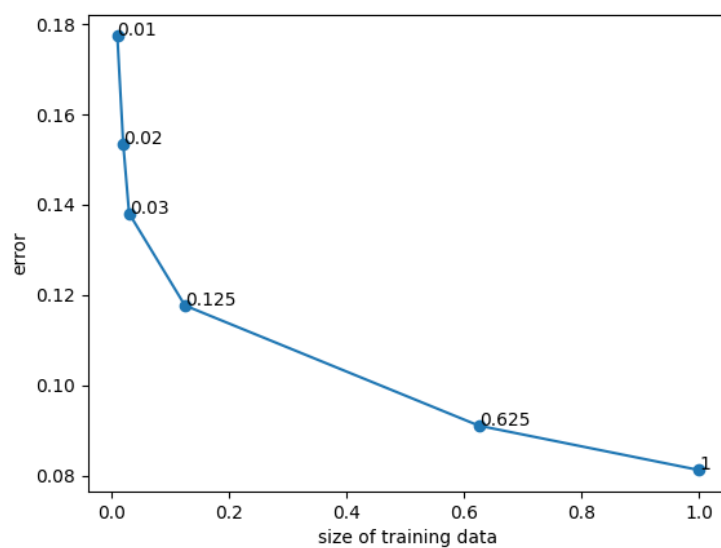


2 Model KNN

Zbiór danych podzieliłam wybierając dane do zbioru treningowego w ten sam sposób co w powyższym modelu. Podczas predykcji wyniku dla danej ze zbioru testowego obliczałam odległość (odległość w rozumieniu przestrzeni euklidesowej) od wszystkich danych ze zbioru treningowego. Wyznaczyłam 16 najbliższych sąsiadów. Jeśli większość z nich miała label 1 to danej ze zbioru testowego też przypisywałam 1, analogicznie dla -1. To, że wybrałam 16 sąsiadów też dobrałam ręcznie. Wyniki uśredniłam dla 3 przebiegów algorytmu. Błędy klasyfikacyjne wynoszą odpowiednio:

0.1774, 0.1535, 0.1381, 0.1177, 0.0910, 0.0812

Poniższy wykres przedstawia krzywą błędu na partycjach zbioru treningowego.

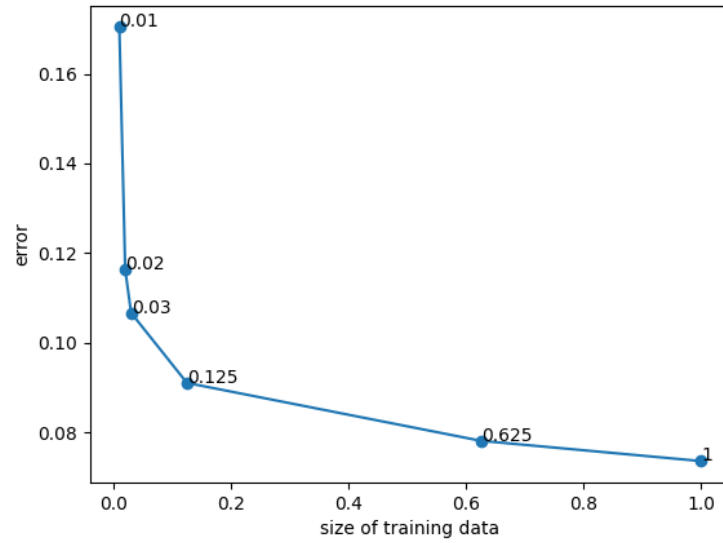


3 Model sieci neuronowej

Zbiór danych podzieliłam wybierając dane do zbioru treningowego w ten sam sposób co w powyższych modelach. Zamieniłam labelle z $-1, 1$ na $0, 1$, wynika to z używania w modelu funkcji aktywacji sigmoid. Cały model składa się z trzech warstw, fully-connected. Pierwsza warstwa ma 30 neuronów, czyli tyle co cech. Druga warstwa ma 15 i trzecia 1 - przez klasyfikację binarną. Wyniki uśredniłam dla 5 przebiegów algorytmu. Błędy klasyfikacyjne wynoszą odpowiednio:

0.1704, 0.1163, 0.1067, 0.0911, 0.0782, 0.0737

Poniższy wykres przedstawia krzywą błędu na partycjach zbioru treningowego.



4 Wnioski

Najszybciej nauczył się model SVM. Z kolei KNN działa zauważalnie gorzej na małych częściach zbioru treningowego. Ostatecznie wszystkie algorytmy osiągały podobne rezultaty 91% – 93% skuteczności.