**Metoda celor mai apropiați vecini**

|  |
| --- |
| Exemplu care arată diferențele dintre metoda celui mai apropiat vecin și metoda celor mai apropiați cinci vecini. Zona colorată reprezintă regiunea de decizie a clasificatorului folosind distanța L2. Se observă că în cazul metodei celui mai apropiat vecin se formează mici ‘insule’ ce pot duce la predicții incorecte. Zonele gri din imaginea 5-NN reprezintă zone de predicție ambigue din cauza egalității voturilor celor mai apropiați vecini. |

În acest laborator vom clasifica cifrele scrise de mână din subsetul **MNIST** folosind metoda celor mai apropiați vecini.

***Descărcați arhiva cu datele de antrenare și testare*** [***de aici.***](https://fmi-unibuc-ia.github.io/ia/Data/data_MNIST.zip)

❓ Care este acuratețea metodei *celui* mai apropiat vecin pe mulțimea de *antrenare* când se folosește distanța L2? Dar pentru distanța L1?

❓ Care este acuratețea metodei celor mai apropiați vecini pe mulțimea de *antrenare* când se folosește numărul de vecini și distanța L2? Dar pentru distanța L1?

**Exerciții**

1. Creați clasa KnnClassifier, având constructorul următor:

|  |
| --- |
| **def** **\_\_init\_\_**(self, train\_images, train\_labels):  self.train\_images = train\_images  self.train\_labels = train\_labels |

1. Definiți metoda *classify\_image(self, test\_image, num\_neighbors = 3, metric = 'l2')* care clasifică imaginea *test\_image* cu metoda celor mai apropiați vecini, numărul vecinilor este stabilit de parametru *num\_neighbors*, iar distanța poate fi L1 sau L2, în funcție de parametrul *metric*.

**Obs:**

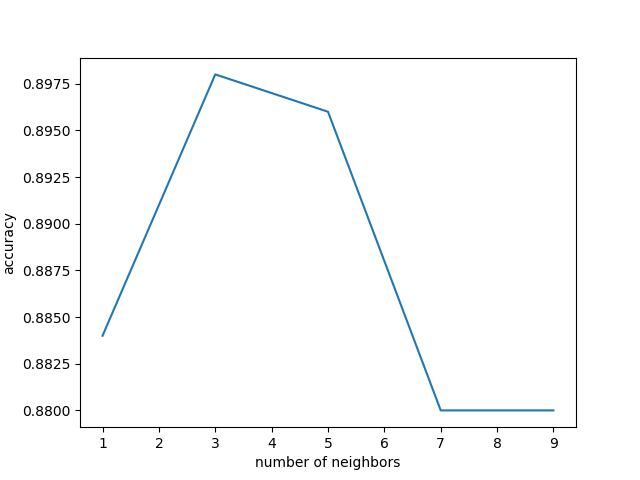
* + În variabilele *train\_images* și *test\_image* valorile unui exemplu sunt stocate pe linie. (train\_images.shape = (num\_samples, num\_features), test\_image.shape = (1, num\_features) )

1. Calculați acuratețea metodei celor mai apropiați vecini pe mulțimea de testare având ca distanța ‘l2’ și numărul de vecini 3. Salvați predicțiile în fișierul *predictii\_3nn\_l2\_mnist.txt*.

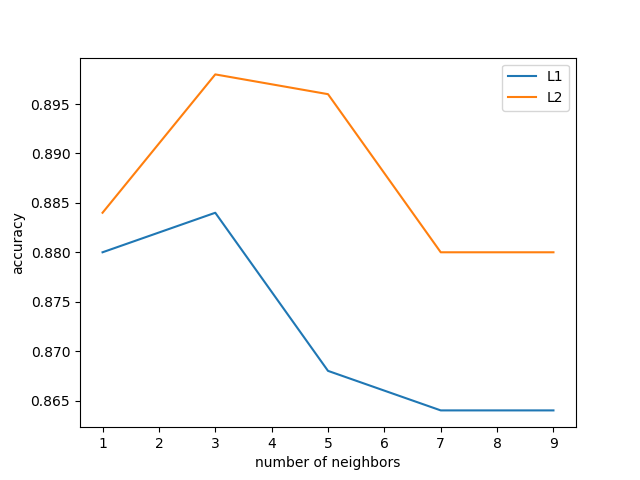
**Obs:**

* Acuratețea pe mulțimea de testare este de 89.8%.

1. Calculați acuratețea metodei celor mai apropiați vecini pe mulțimea de testare având ca distanța L2 și numărul de vecini [1, 3, 5, 7, 9].
   1. Plotați un grafic cu acuratețea obținuta pentru fiecare vecin și salvați scorurile în fișierul *acuratete\_l2.txt*.



* 1. Repetați punctul anterior pentru distanța L1. Plotați graficul de la punctul anterior în aceeași figură cu graficul curent (utilizați fișierul *acuratete\_l2.txt*).



Funcții numpy:

|  |
| --- |
| np.sort(x) *# sorteaza array-ul*  np.argsort(x) *# returneaza indecsi care sorteaza array-ul* np.bincount(x) *# calculeaza numarul de aparatii al fiecarei valori din array*  print(np.bincount(numpy.array([0, 1, 1, 3, 2, 1, 7]))) *# array([1, 3, 1, 1, 0, 0, 0, 1])*  np.where(x == 3) *# returneaza indecsi care satisfac conditia*  np.intersect1d(x, y) *# returneaza intersectia celor 2 array*  np.savetxt(‘fisier.txt’, y) *# salveaza array-ul y in fisierul fisier.txt* |