kNearestNeighbors: Using Your Algorithm to Do More

Congratulations on finishing your kNearestNeighbors classifier! You’re now a programmer for social good. This project has hopefully started showing you how using your knowledge from CS, **you can be a changemaker**.

**Extensions**

It’s time to explore the material in more detail. You can pursue any of the options below to become even more of a changemaker or learn even more about technology, all while getting some extra credit! We’ve listed a couple of extension options below, but if you have other ideas for how to explore machine learning/social impact, let your teacher know, and they will tell you whether they’ll consider it for extra credit.

**Option One**: Use your machine learning algorithm on other data. A couple of links to datasets are included at the bottom of this document. The core parts of the algorithm will be almost the same as for breast cancer classification, but there are unique technological challenges when getting this to work for other datasets. These include:

* Cleaning the data (formatting it so your algorithm can process it properly), and modifying the InputHandler class to read in new data formats
* Splitting data appropriately into training sets, validation sets, and testing sets. To do this, understand the differences, and figure out why you can’t just have two sets like we did for the first project.
* Improvements to the accuracy of the algorithm -- this is where you can get creative! Some ideas include:
  + Normalization (some features might range from 1-100, but others might range from 0-1, which means the distance will depend a lot more on the first feature).
  + Weighting features (maybe some are more important than others?)
  + Modifying values of K
  + And more!

Trying to train the kNN classifier on other datasets listed below will be an exciting challenge -- and rewarding! They’ll hone your software engineering skills and make you a better programmer for social good.

**Option Two**: Learn more machine learning. The Coursera course on machine learning by Andrew Ng (from Stanford) is the standard introductory ML course. (It does require some familiarity with working with matrices and some familiarity with the concept of derivatives). Take a look at it, and start watching the video lectures.

<https://www.coursera.org/learn/machine-learning>

**Option Three**: If you can think of other creative ways to extend the project, let your teacher know, and have fun!

**Datasets for Option One:**

* Australian Sign Language -- convert sign language (read from glove sensors) to words.
  + <http://archive.ics.uci.edu/ml/datasets/Australian+Sign+Language+signs>
  + <http://archive.ics.uci.edu/ml/machine-learning-databases/auslan2-mld/auslan.data.html>
  + To make this work, you’ll likely need to average over all the frame readings for one instance of a sign.
* Fetal heart rate (Classify as abnormal, normal, pathologic based on other characteristics that are easy to observe)
  + <http://archive.ics.uci.edu/ml/datasets/Cardiotocography>
  + <https://gsm672.wikispaces.com/Cardiotocography+-+Predicting+Fetal+Heart+Rate>
* Parkinson’s — automatic detection of freezing of gait while walking
  + <http://archive.ics.uci.edu/ml/datasets/Daphnet+Freezing+of+Gait>
* Detection of Parkinson’s from audio data
  + <http://archive.ics.uci.edu/ml/datasets/Parkinsons>