Python and k-Nearest Neighbor

Resources:

- https://medium.freecodecamp.org/why-you-need-python-environments-and-how-to-manage-them-with-conda-85f155f4353c
- https://www.youtube.com/watch?v=4HKqjENq9OU
- https://www.datacamp.com/community/tutorials/introduction-machine-learning-python
- https://shapeofdata.wordpress.com/2013/04/23/nearest-neighbors-classification/
- https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

Setting up Python

- Miniconda - contain the conda package manager and Python
- Anaconda – More packages/libraries available (also includes R Studios) and includes GUI

Both include Python and Conda (environment manager) and Pip (package manager)

Installing Miniconda

https://docs.conda.io/en/latest/miniconda.html

- conda list – lists the packages you current have
- conda search packagename
- conda install packagename

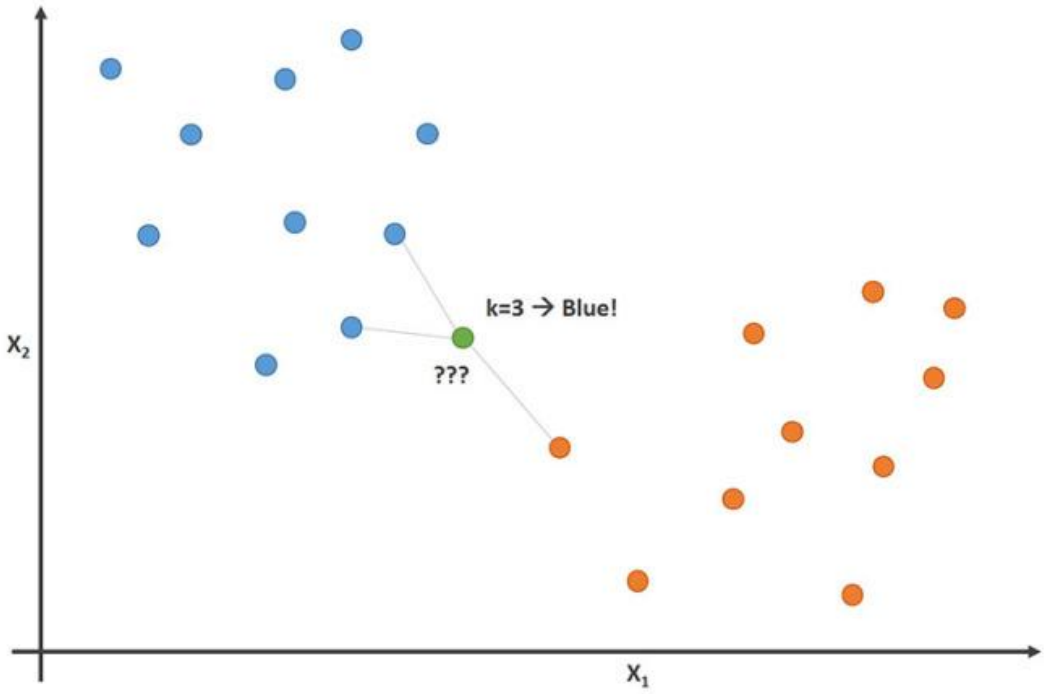
Useful packages

- Sklearn (scikit-learn)
    - Sklearn is machine learning python library. Features various classification, regression and clustering algorithms.
- Pandas (includes Numpy)
    - Pandas provides useful dataframe. Dealing with data in tabular manner.
    - NumPy helps with large, multi-dimensional arrays and matrices. Includes mathematical functions to operate on these arrays
- Matplotlib
    - Has pyplot which plots data like MATLAB

k-Nearest Neighbor

Is a **supervised** learning algorithm. Usually used for **classification** problems but also can be used regression problems.

Very simple overview is:



k = number of "neighbors" you want to include in judging your new data point

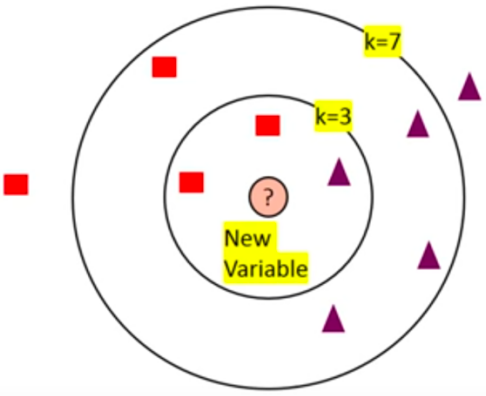
non-parametric: we don't care about statistical probabilities or distributions when classifying new data points.
lazy: doesn't use training data for generalization/modeling. Only computes when you "input" a test data. Saves all training data.

Most common calculation of distance between neighbors uses Euclidean distance:

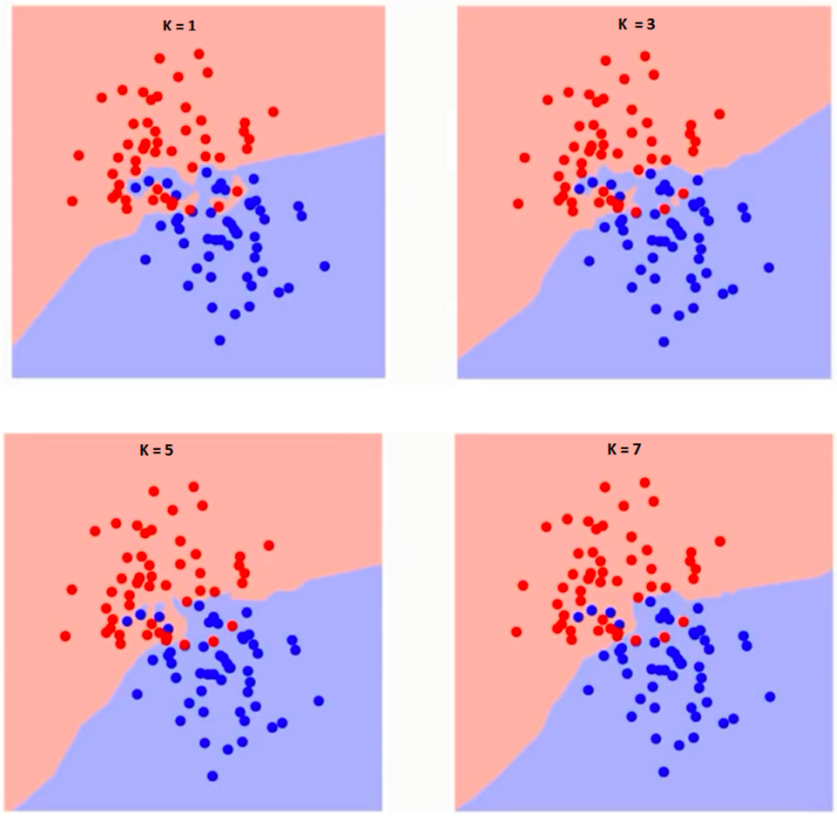
$$d(x, x') = \sqrt{(x_1 - x_1')^2 + (x_2 - x_2')^2 + \ldots + (x_n - x_n')^2}$$

For other calculation methods see: https://www.saedsayad.com/k_nearest_neighbors.htm

How to pick a value for k



- Can sample your training data to assess what's a good number
    - Address the sample of training data as unknown and see how different values of k predict the training data.
- For 2 class variable: k should be odd number.
    - (k must not be a multiple of the number of variable → to avoid ties)
- Lower values of k can be noisy. Can run into chance of overfitting the data. (think outliers in training data can skew your prediction)
- Larger values of k may cause oversimplifying differences between groups. Also takes longer to compute.
- $\sqrt{n}$ is a common k value.



When kNN is used

- Data is noise free and accurately labeled.
- Initial overview of data for preliminary results.
- Recommender systems
- Large data set means higher accuracy but because of the “lazy” characteristic of the algorithm, storage and computation power may be an issue.
- No long training phase.