Nearest Neighbors

COSC 410: Applied Machine Learning

Spring 2022

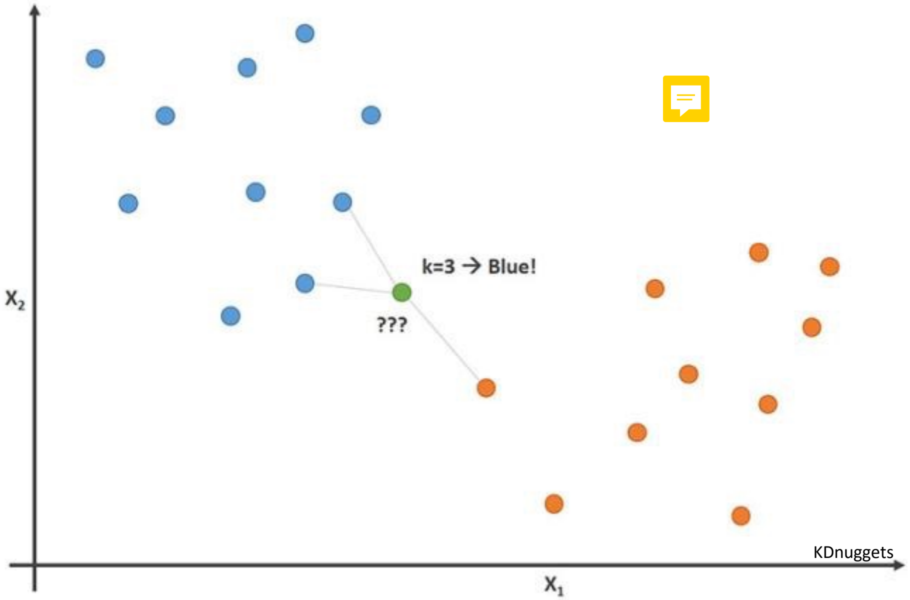
Prof. Apthorpe

Outline

- Nearest Neighbors
  - Classification & Regression
  - Training
  - Performance
- Programming Practice

K-Nearest Neighbors

- Predict the mean or mode label of the *k* closest* examples in the training set
  - Use distance function of your choice to define “closest”
  - Use **mode** for classification, **mean** for regression



KDnuggets

KNN: All *k* nearest neighbors count **equally** in mode or mean calculations

Weighted KNN: Weight contribution of k nearest neighbors based on distance

Feature standardization typically required! Why?

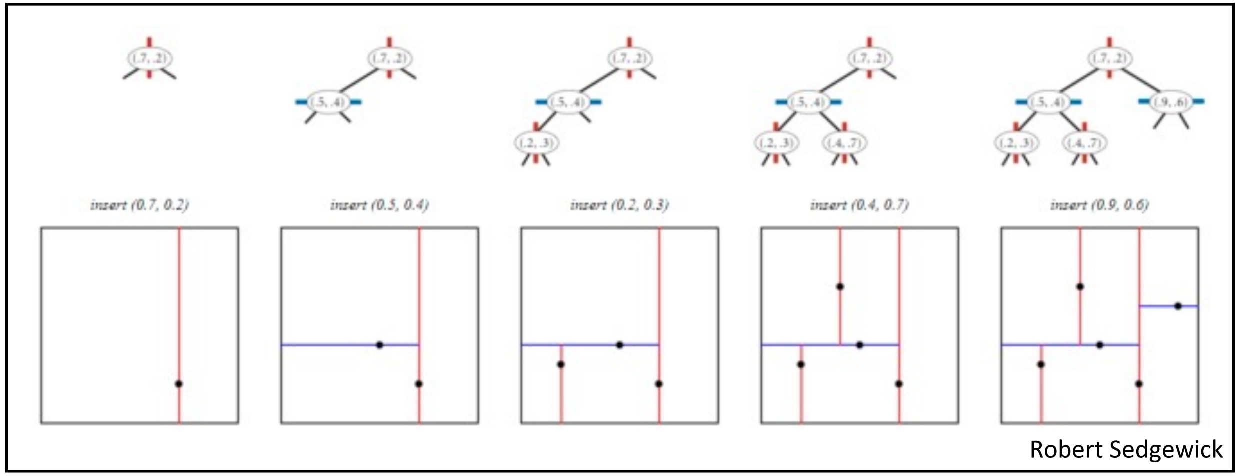
K-Nearest Neighbors Training

- Recall from last class:
  - To train a model, we **find model parameters that minimize the error function** given the training data, often using gradient descent

- **KNN is a 0-parameter model!!**
  - Training is just “store the training data” → No GD or other optimization required
  - *k* and the choice of distance function are **hyperparameters**
    - Try several options and compare performance with cross-validation

K-Nearest Neighbors Performance

- KNN **computational performance** scales poorly with increasing **# of examples**
  - More possible neighbors that need to be checked
  - More storage required for “trained” model
  - Can use KD-tree encodings of data to optimize neighbor-finding



Robert Sedgewick

K-Nearest Neighbors Performance

- KNN **prediction performance** scales poorly with increasing **# of features**

- Many distance metrics don't work well in high dimensions
  - The **"curse of dimensionality"**
  - *"In high dimensional space, everything is far away"*

- Scikit-Learn has many distance functions tailored for different types of data (Boolean, integer, etc.), but none avoid the curse of dimensionality

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.DistanceMetric.html

Programming Practice

RegressionKNN.ipynb

Questions?