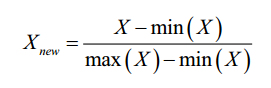
* What is knn
  + Nn used for
    - optical character recognition, faicial recognition
    - Netflix Algorithm
    - Identifying patterns in genetic data, e.g. proteins and diseases
  + Good for situation where there is a very complicated underlying model, but with relatively distinct classes.
  + Simple, no assumptions about underlying data, and fast to train
  + Does not produce a model, slow to classify, large space requirement, and extra processing required for nominal /missing data
* Calculating the nn
  + Calculating the distance
    - Euclidian distance
      * Sqrt(sum of squared differences)
* Choosing how many nn, e.g. k
  + Bias variance tradeoff
    - Larger k reduces variance due to noise
    - But run risk of overgeneralization, missing small patterns
    - Small k may be overfit.
  + In practice, depends on the difficulty of the concept to be learned & size of the data
  + Common range is 3 to 10
  + Could use sqrt of the input size
  + Usually test several k
  + Can increase k as data size increases
    - Even the rare, but valid, trends will have enough nearest neighbors to classify
* Preparation for use with kNN
  + Might need to rescale variables
    - E.g. When comparing foods, sweetness is measured 0-10 but spiciness may be measured by Scoville units, 0 to >1,000,000
  + Methods to rescale
    - Traditional method: min-max normalization
      * all features transformed into range 0-1
      * essentially, subtract minimum of feature X from each value and divides by the range of X
      * 
    - Z-Score standardization
      * X minus mu over sigma
    - Make 0-1 for nominal data
      * For n category, just make a variable 1 or 0 like a flag
      * For last, leave out, as in default case
* Why is kNN lazy?
  + No abstraction occurs, nor generalization
  + Don’t really learn anything, just storing training data verbatim.
  + Quick to train, slow to classify
  + Aka instance-based or rote learning
* Diagnosing breast cancer with kNN