Artificial Intelligence

KNN

K-Nearest-Neighbors

KNN ?

- Supervised Machine-Learning algorithm
  - Can be used for both classification and regression
- "Instance-based" learning
  - Uses training instances to directly make predictions
- Relies on a distance metric
  - Manhattan distance
  - Euclidian distance (most common)
  - Minkowski distance (generalizes the previous)
  - Hamming distance (# differences at the same index)
    - For categorical features, properly encoded

KNN - Algorithm

- To classify a new sample
  - Measure its distance to ALL the samples in the dataset
  - Sort the distances in ascending order
  - Select the neighbors: the K first samples of the sorted result
  - For classification, assign the new sample the most common class ("majority voting")
    - Or use weighted voting
    - Or pick randomly
  - For regression, average the target values of the neighbors

KNN

- Non-parametric
  - No assumptions about the data distribution
- Lazy learning
  - Not exactly a model computation
  - More a dataset memorization, to be used at prediction time
    - Instant fit
    - Slower predict
- Versatile
  - Classification
    - The voting of the K-Nearest-Neighbors
  - Regression
    - The average of the numerical targets of the K-Nearest-Neighbors
      - Average, or other aggregate measure
  - Search
    - Recommender systems

KNN

- Value of K?
  - Small
    - Overfitting?
  - Big
    - Underfitting?
- Distance metric? Depends on the data
- Feature scaling?
  - It benefits from normalization and standardization
  - Normalization (SciKit's MinMaxScaler)
    - To scale the data to a fixed range (e.g. [0,1])
      - N(x) = x - min(X) / max(X) - min(X)
      - x is a sample and X the entire dataset ; operations are performed per feature
  - Standardization (SciKit's StandardScaler)
    - Mean = 0 ; Stdevp = 1
      - S(x) = x - Mean(X) / Stdevp(X)
      - x is a sample and X the entire dataset ; operations are performed per feature
- Many features?
  - Potential performance problem

KNN - sample code

- At: https://github.com/amsm/am_knn
- There is code for
  - a procedural implementation of KNN
    - applied to Ronald Fisher's Iris dataset
  - an OOP implementation of KNN
    - applied to an artificial dataset
  - some distance metrics
    - the Euclidian distance
    - the Minkowski distance
- And, considering this non-parametric algorithm's extreme sensitivity to the features' values, also implementations of
  - Normalization of features
  - Standardization of features

References

- https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
- https://github.com/amsm/am_knn