K NEAREST NEIGHBOR CLASSIFIER

DR. FARHAD RAZAVI

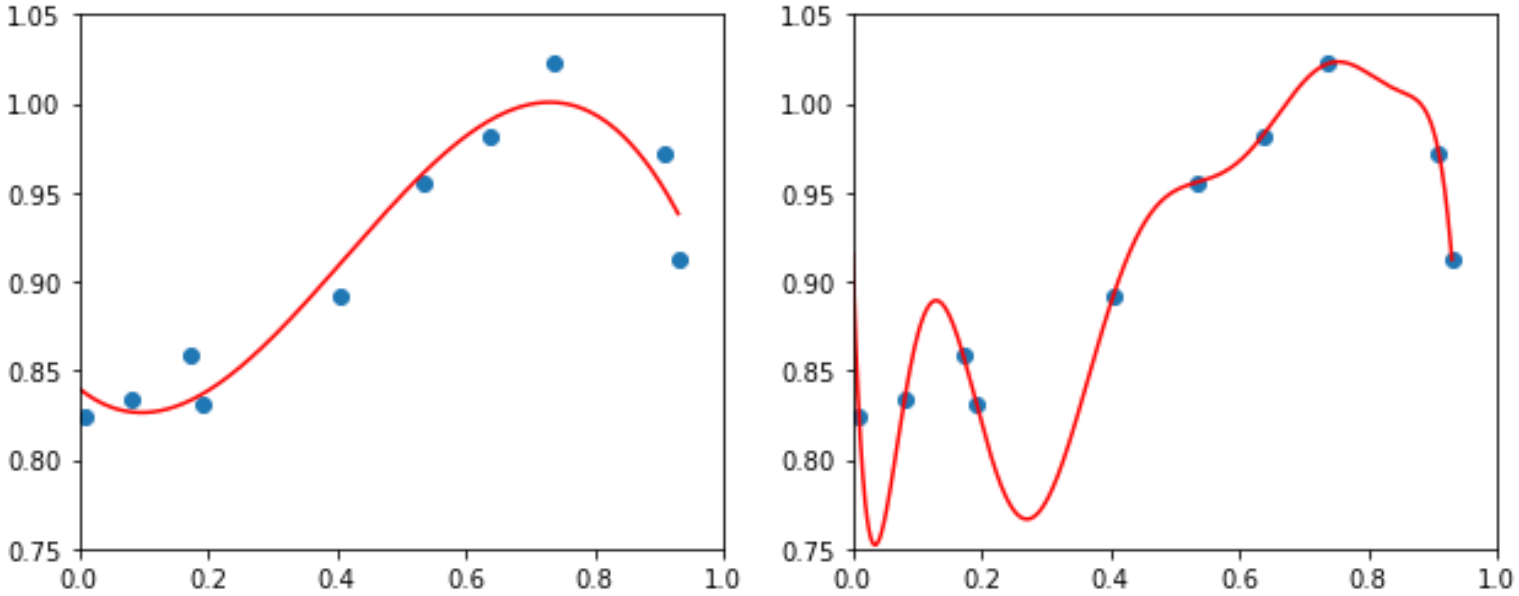
Based on Professor Alexander Ihler material on this subject

OUTLINE

- Complexity and Overfitting
- Cross-validation technique
- Nearest Neighbor
- K-Nearest Neighbor (KNN)
  - Regression model
  - Classification model

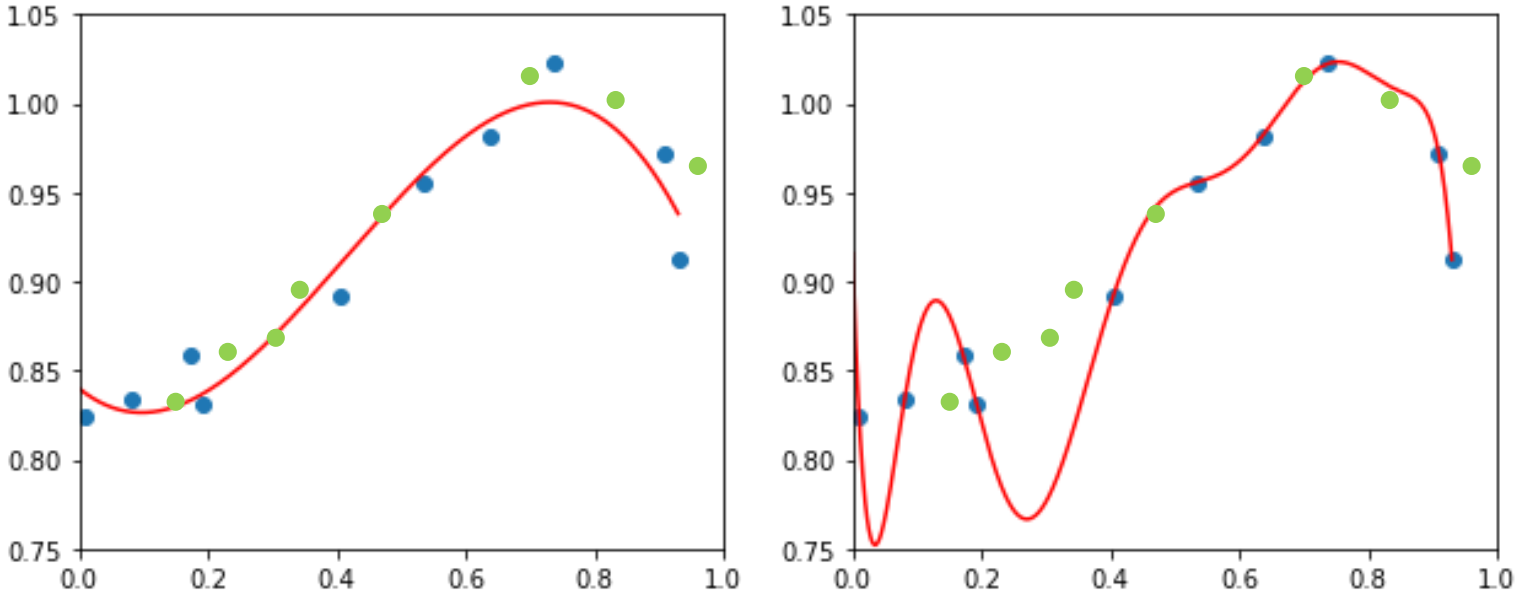
OVERFITTING AND COMPLEXITY

- How can we infer we have overfitted our data?



OVERFITTING AND COMPLEXITY

- One way was to add more data! What if we do not have any means to collect more data?!



DIVIDING THE DATASET

- One effective way to evaluate our model against the errors of high bias and high variance is to randomly divide the entire dataset to two sets of "training data" and "test data".

- Learn the parameters of the model from training data.
- Vary the model complexity and evaluate it against test data and pick best performer.

- Problem:
  - Over-estimates test performance ("lucky" model).
  - Learning algorithms should never have access to test data.

TRAINING, VALIDATION, AND TEST DATA

- One other way is to reserve some data for validation only.

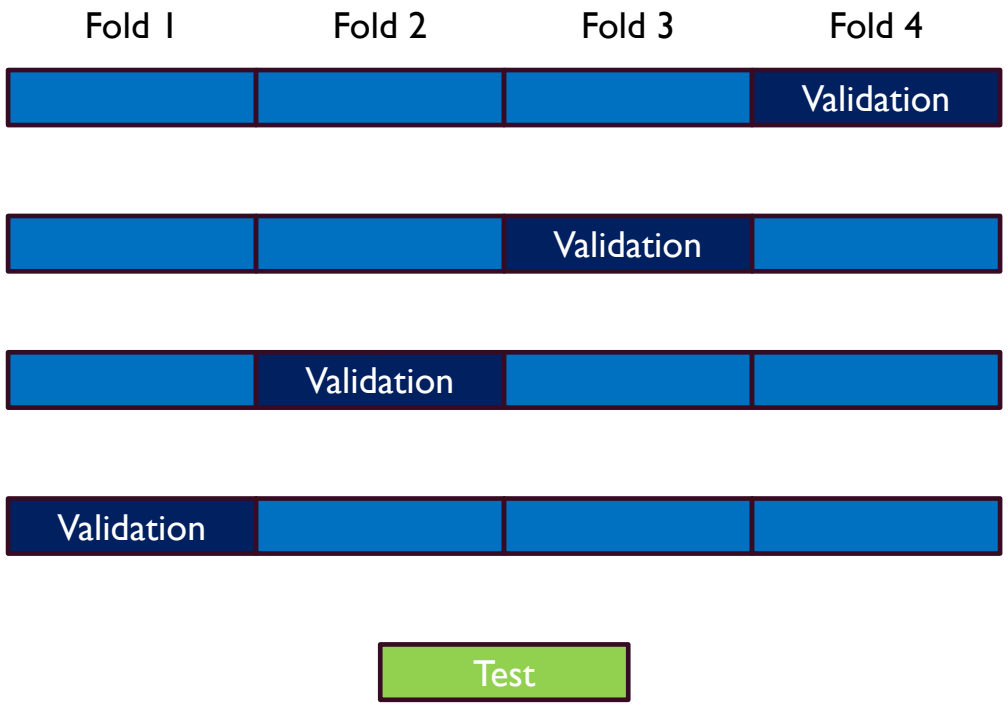
- Learn parameters of each model from training data
- Evaluate models on validation data, pick best performer.
- Reserve test data to benchmark chosen model.

- Problem:
  - Wasteful of training data (learning can't use validation).
  - May bias selection towards overly simple models.

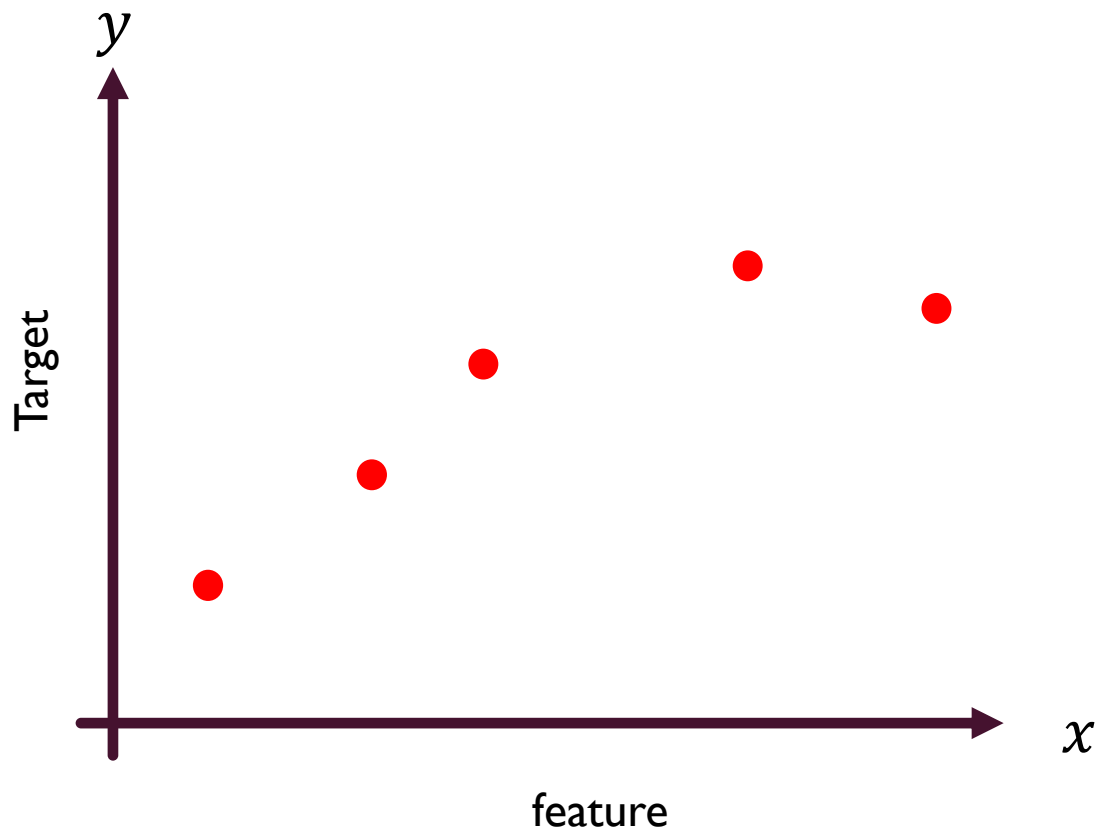
CROSS-VALIDATION

- Divide training data into K equal-size folds.
- Train on K-1 folds, evaluate on remainder.
- Pick model with best average performance across K-trials.

- How many folds?
  - Bias: Too few, and effective training dataset much smaller
  - Variance: Too many, and test performance estimates noisy
  - Cost: Must run training algorithm once per fold (parallelizable)
  - Practical rule of thumb: 5-fold or 10-fold cross-validation

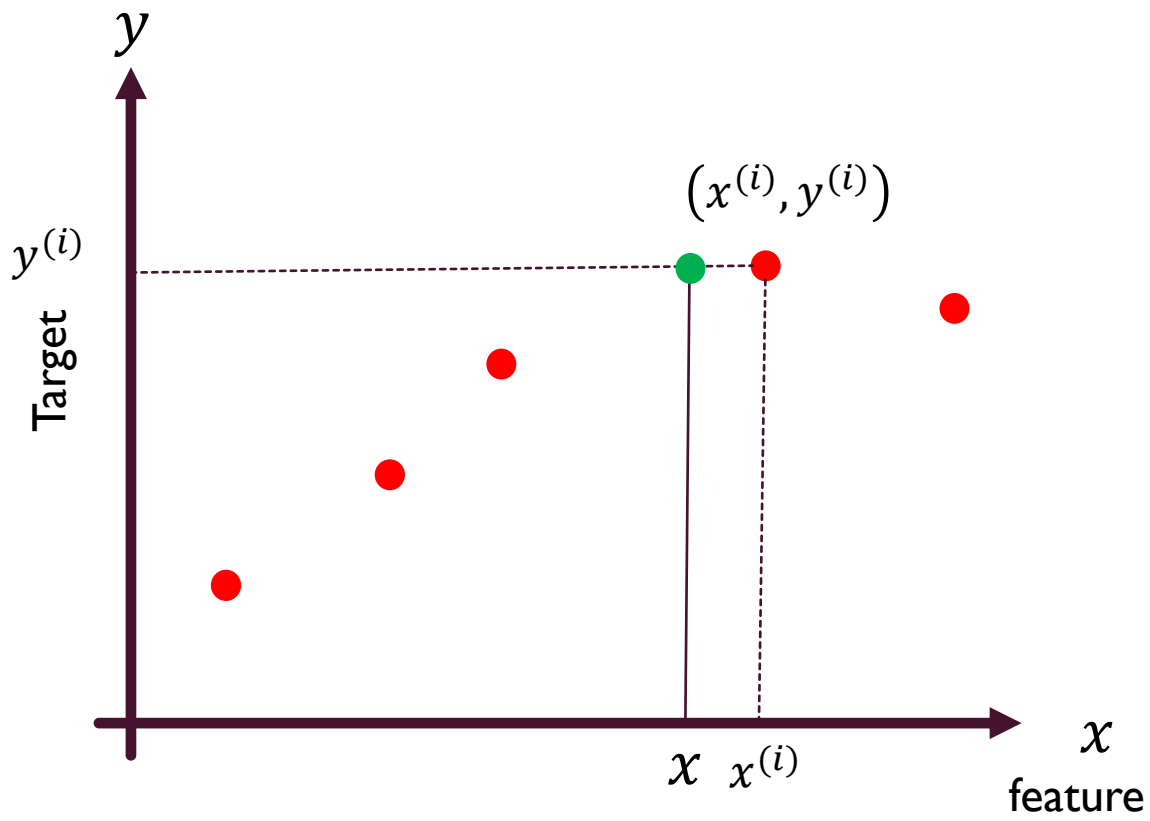


NEAREST NEIGHBOR



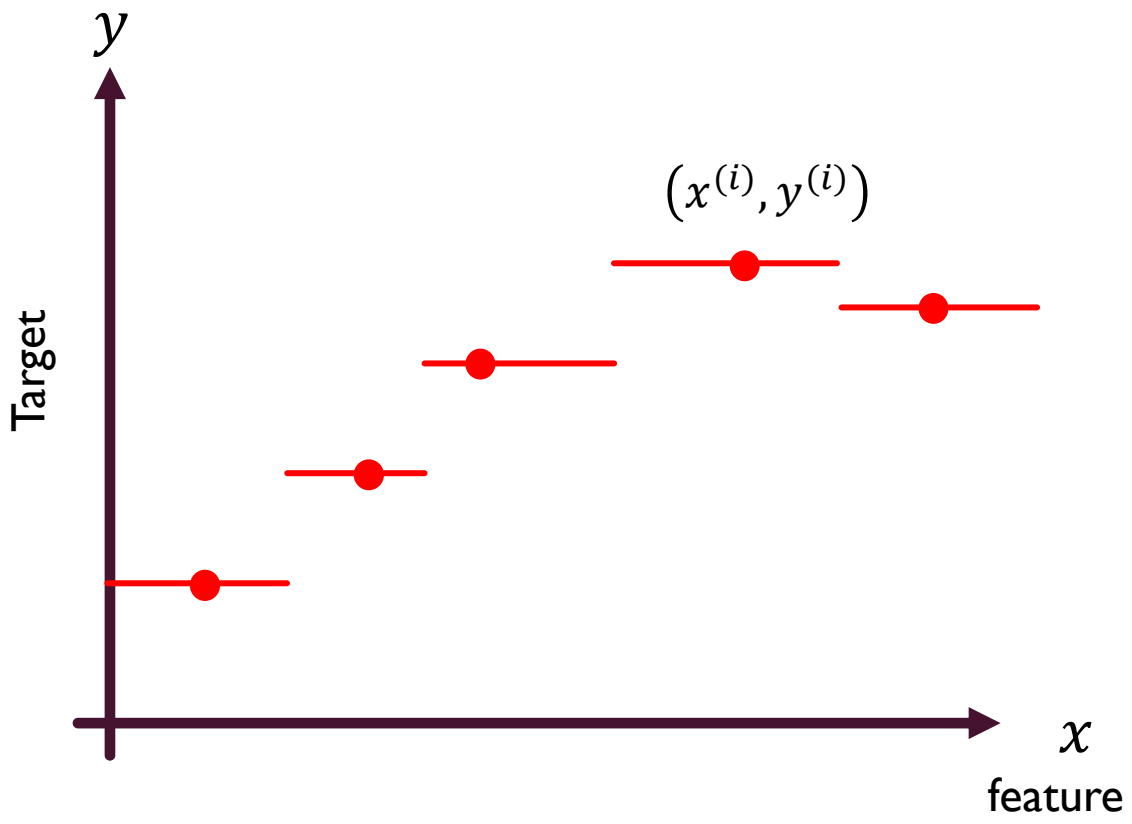
- We have already seen linear regression and its modified forms which could handle nonlinear models.
- Adding nonlinear features inherently made the model more complex,
- A big issue was the high variance when the model was getting more complex.

NEAREST NEIGHBOR



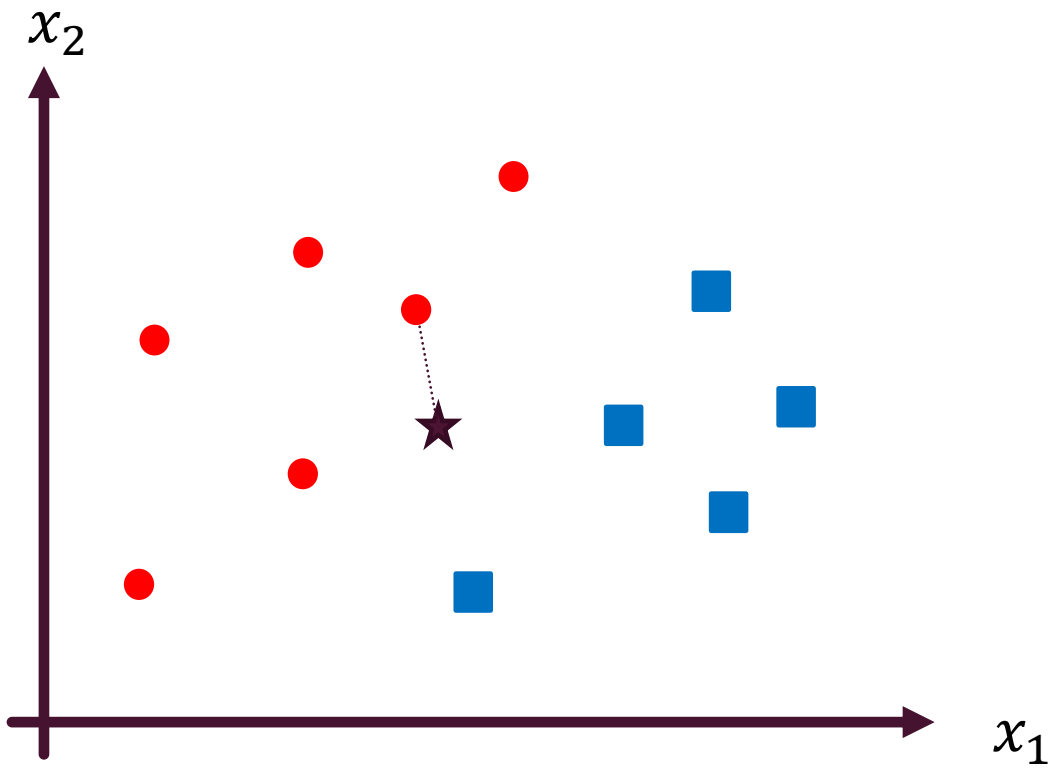
- What is the simplest form of prediction based on current data?
- It is intuitive to assume the value of the model (function) has not changed much in the vicinity of the current known training values of $(x^{(i)}, y^{(i)})$.
- For a new feature x, Nearest Neighbor predicts $\hat{y}$ to be the target value $y^{(i)}$ of the closes training pair $(x^{(i)}, y^{(i)})$.

NEAREST NEIGHBOR REGRESSION



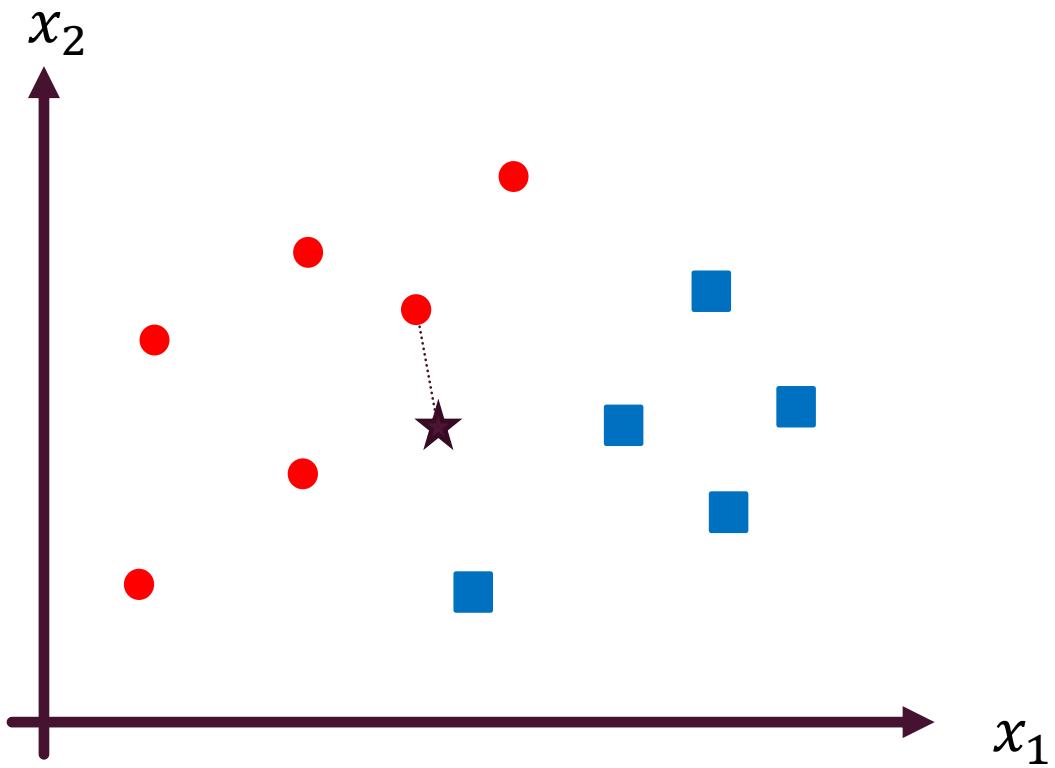
- Now our predictive model will find the closest training datum to our new feature and return the target value of that training datum.
- This will form a piecewise constant model $f(x)$.
- Notice that there is no parameter to be found unlike all the models we have discussed so far $(f_{w,b}(x))$.

NEAREST NEIGHBOR CLASSIFIER



- In the case of a classifier similarly the predictive model will find the closes training datum and assign its class to the new feature.
- Notice that unlike logistic regression we do not assign any confidence value to the prediction. (there is no probability value assigned).

NEAREST NEIGHBOR CLASSIFIER



- In the case of a classifier similarly the predictive model will find the closest training datum and assign its class to the new feature.
- Notice that unlike logistic regression we do not assign any confidence value to the prediction. (there is no probability value assigned).

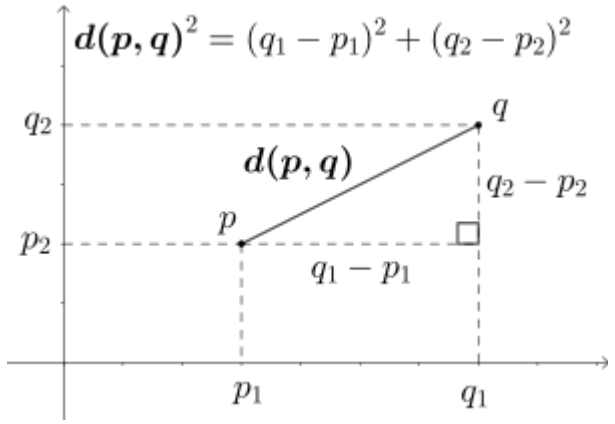
DISTANCE DEFINITION

- In Nearest Neighbor we need to find the closest training data point.
- This requires a way to define the distance between our data points and our new feature.
- Distance definition:
  - Euclidean distance
  - Manhattan distance
  - Hamming distance
  - …

EUCLIDEAN DISTANCE

- Euclidean distance between two points in Euclidean space is the length of a line segment between the two points.
- It can be calculated from the Cartesian coordinates of the points using the Pythagorean theorem.
- One dimension:

$$d(p, q) = |p - q|.$$
$$d(p, q) = \sqrt{(p - q)^2}.$$



- Two dimension:

$$d(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}.$$

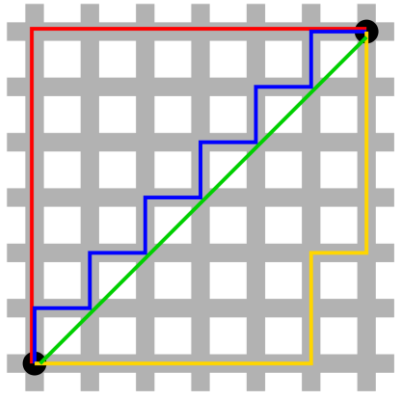
- Multi-dimension:

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_i - q_i)^2 + \cdots + (p_n - q_n)^2}.$$

MANHATTAN DISTANCE

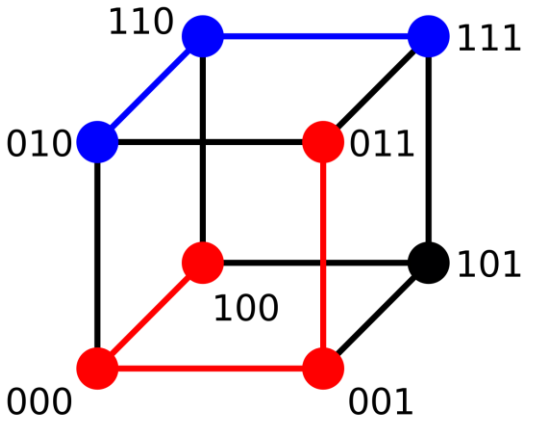
- The distance between two points is the sum of the absolute differences of their Cartesian coordinates.
- It is also known as taxicab metric, rectilinear distance, L_1 distance, L^1 distance, l_1 norm, city block distance, snake distance.

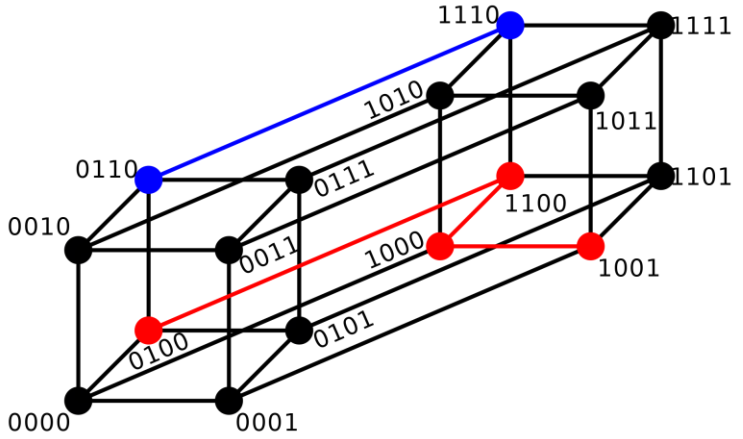
$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|,$$



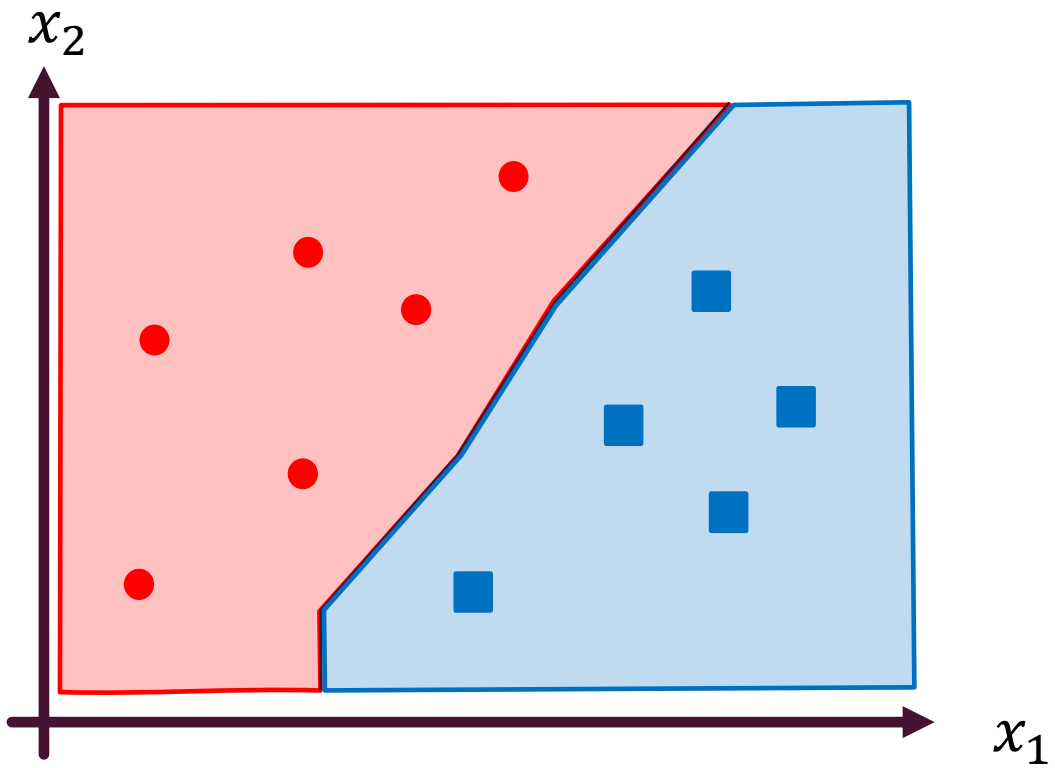
HAMMING DISTANCE

- The Hamming distance between two equal-length strings of symbols is the number of positions at which the corresponding symbols are different.
- The symbols may be letters, bits, or decimal digits, among other possibilities. For example, the Hamming distance between:
  - Examples:
    - Hamming distance between "karolin" and "kathrin" is 3.
    - Hamming distance between "karolin" and "kerstin" is 3.
    - Hamming distance between "kathrin" and "kerstin" is 4.
    - Hamming distance between 0000 and 1111 is 4.
    - Hamming distance between 2173896 and 2233796 is 3.



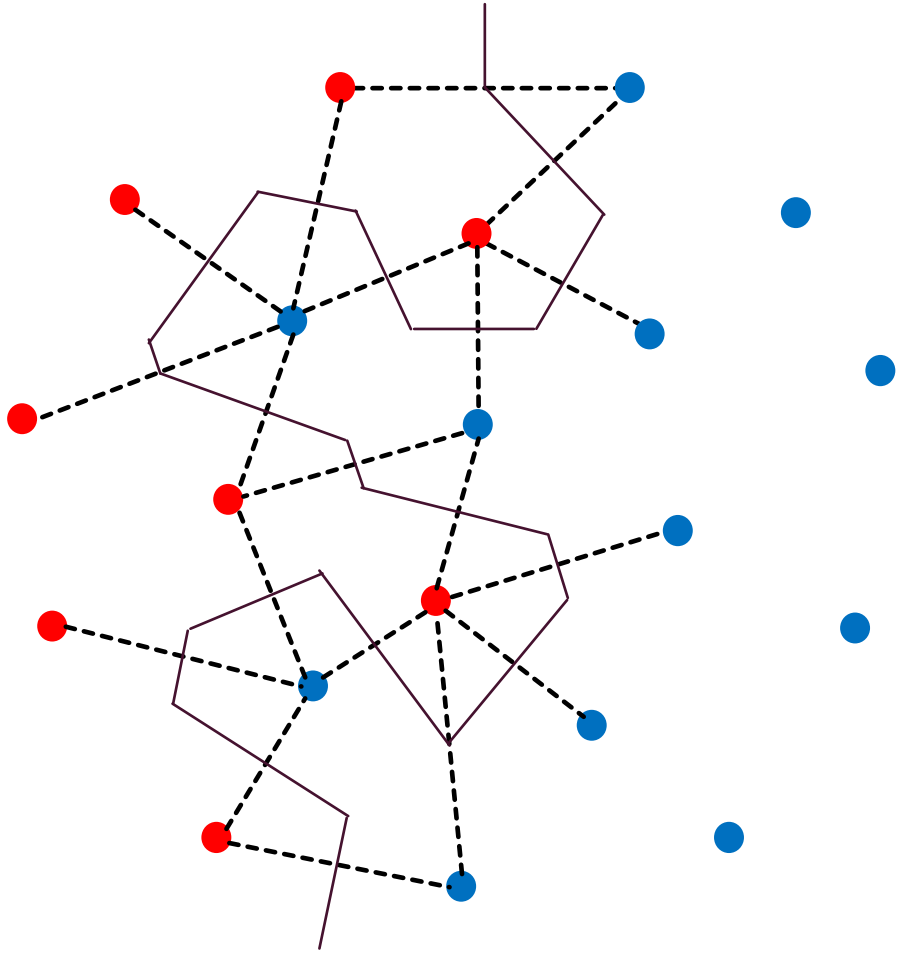


NEAREST NEIGHBOR CLASSIFIER



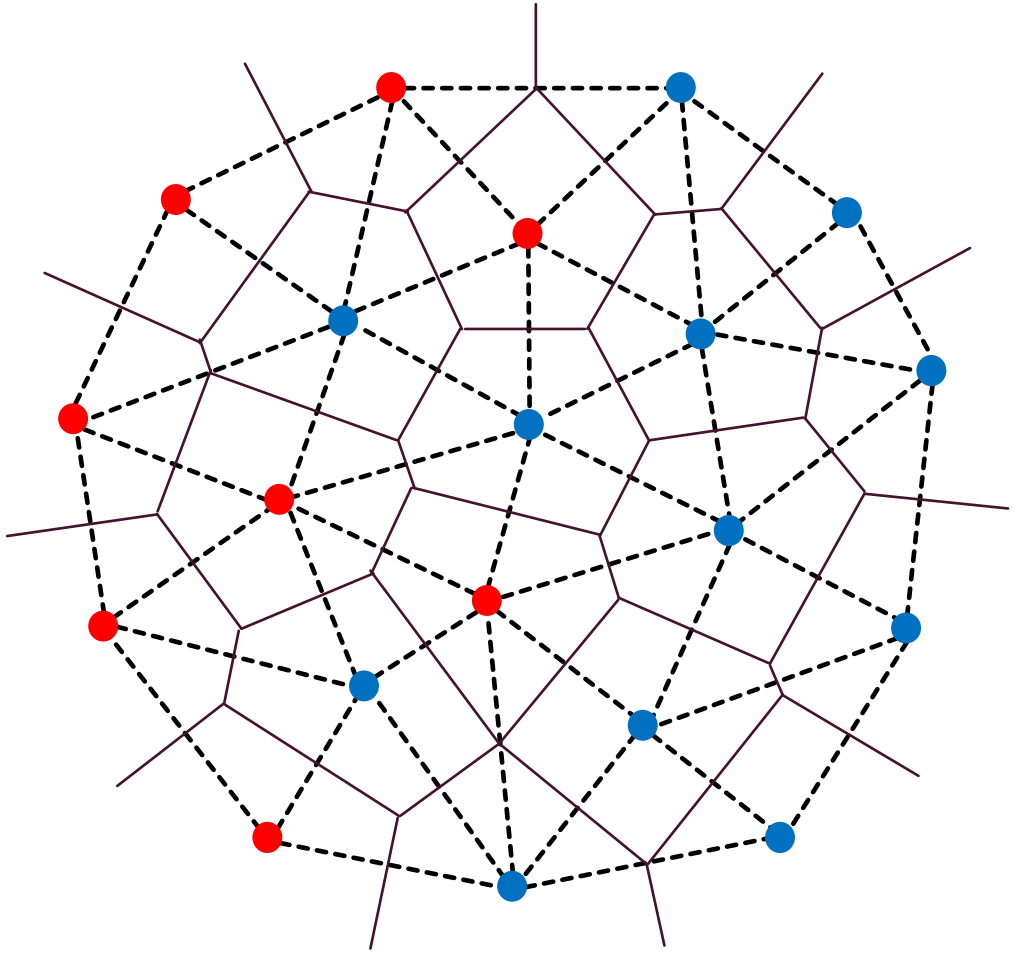
- Decision boundary will be piecewise linear.

VORONOI DIAGRAM AND DELAUNAY TRIANGULATION



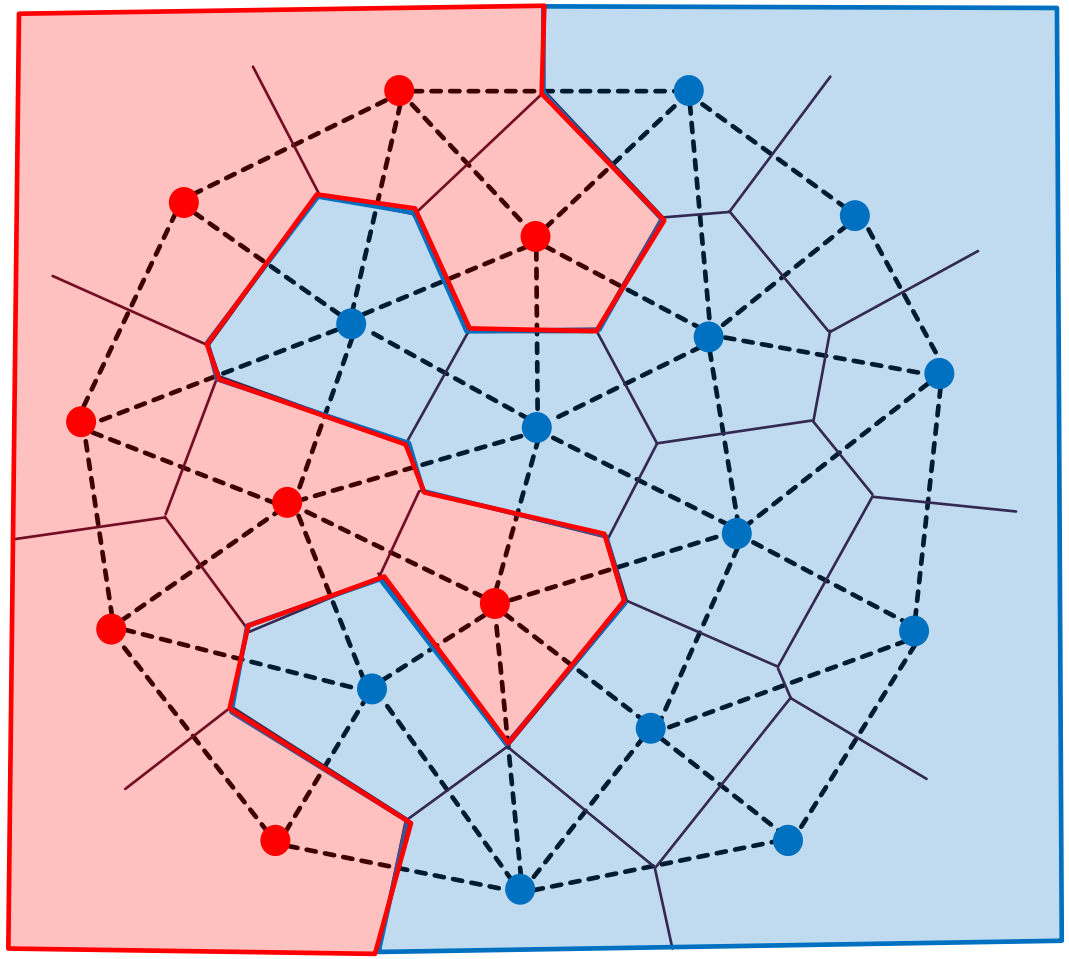
- Voronoi diagram:
  - Each data point is assigned to a region, in which all points are closer to it than any other data point.
- Delaunay triangulation:
  - Dual graph of Voronoi diagram
- Decision boundary:
  - Those edges across which the decision changes
  - They are the centers of the circumcircles of Delaunay triangulation.

VORONOI DIAGRAM AND DELAUNAY TRIANGULATION



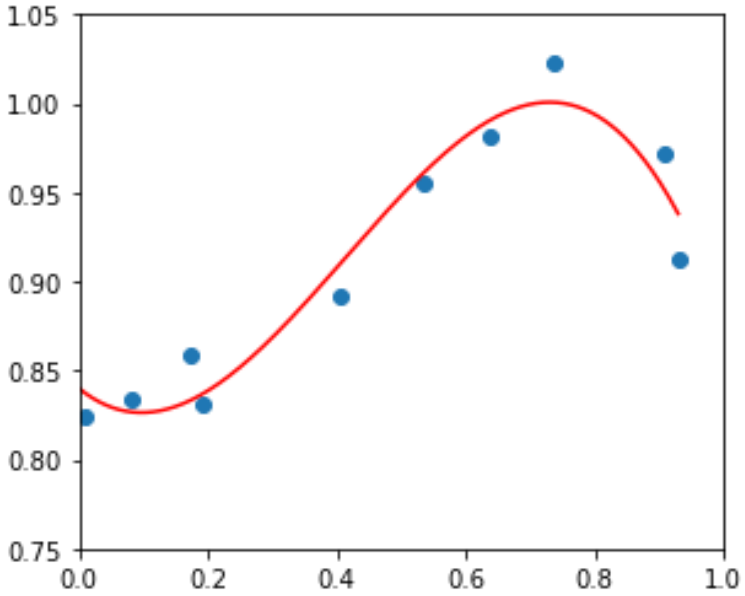
- Voronoi diagram:
  - Each data point is assigned to a region, in which all points are closer to it than any other data point.
- Delaunay triangulation:
  - Dual graph of Voronoi diagram
- Decision boundary:
  - Those edges across which the decision changes
  - They are the centers of the circumcircles of Delaunay triangulation.

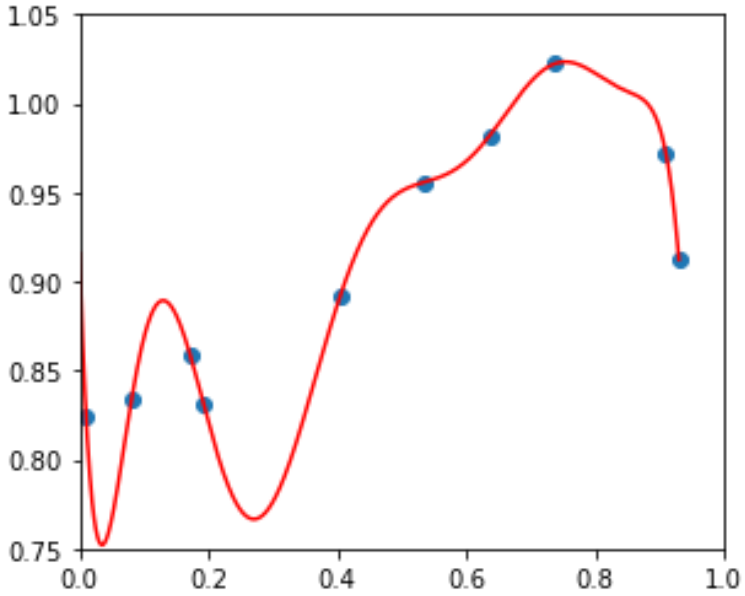
VORONOI DIAGRAM AND DELAUNAY TRIANGULATION

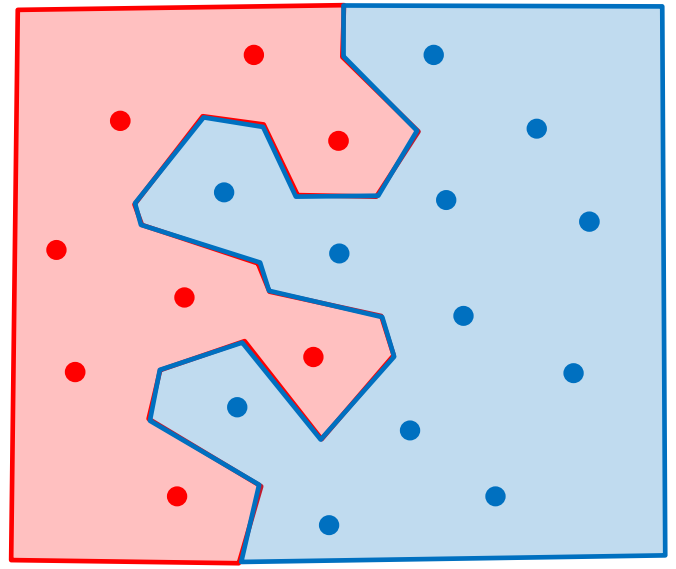


- Voronoi diagram:
  - Each data point is assigned to a region, in which all points are closer to it than any other data point.
- Delaunay triangulation:
  - Dual graph of Voronoi diagram
- Decision boundary:
  - Those edges across which the decision changes
  - They are the centers of the circumcircles of Delaunay triangulation.

ISSUE OF COMPLEX BOUNDARY







- Even though Nearest Neighbor (1NN) is a very simple model, the decision boundary tend to be complex.
- This means the model have a low bias but high variance.

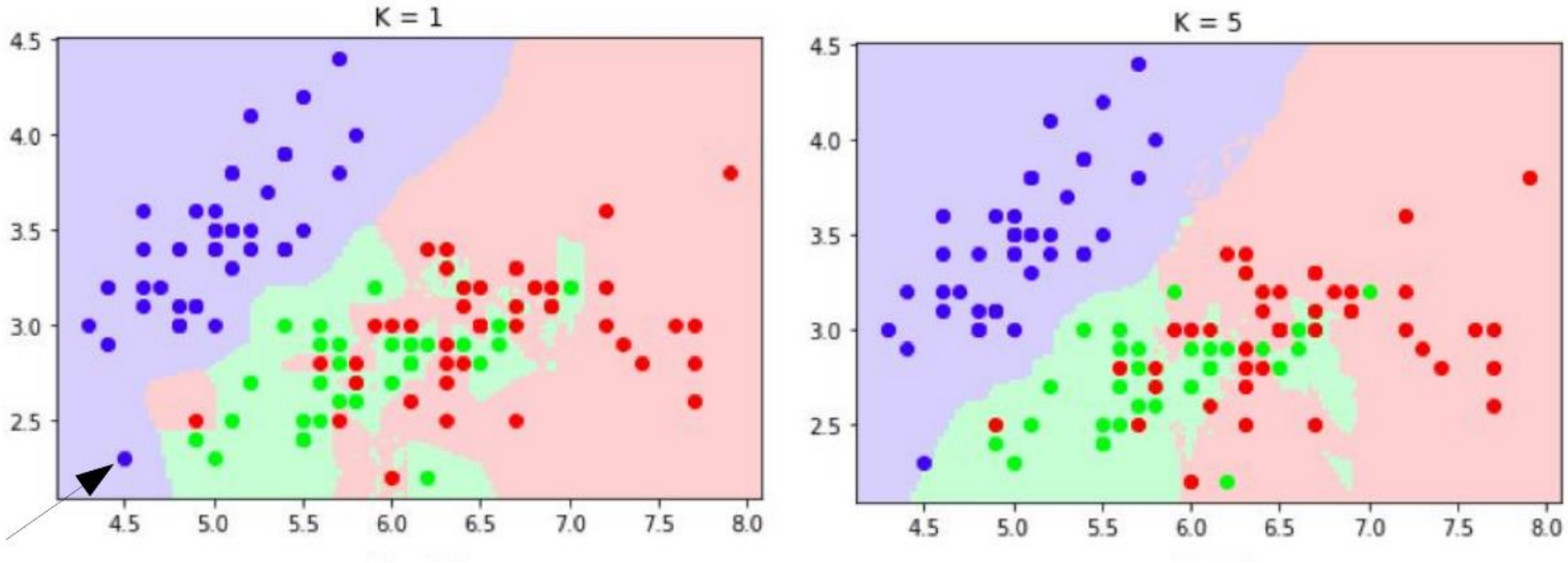
K-NEAREST NEIGHBOR (KNN) CLASSIFIER

- Find the k-nearest neighbor to $\vec{x}$ in the data
  - Rank the feature vectors according to Euclidean distance
  - Select the k vectors which have smallest distance to $\vec{x}$.
- Regression
  - Usually just average the y-values of the k closest training examples
- Classification
  - Ranking yields k feature vectors and a set of k class labels.
  - Pick the class label which is most common in this set ("vote")
  - Classify $\vec{x}$ as belonging to this class
  - Note: for two-class problems, if k is odd (k=1, 3, 5, …) there will be no "ties", otherwise, just use (any) tie-breaking rule.
- Training is trivial: just use training data as a lookup table (memorizes everything), and search to classify a new datum. Because no learning is happening KNN is often referred to as a lazy learner among practitioners!
  - Can be computationally expensive (must fund nearest neighbors within a potentially large training set).

Lazy Learner!

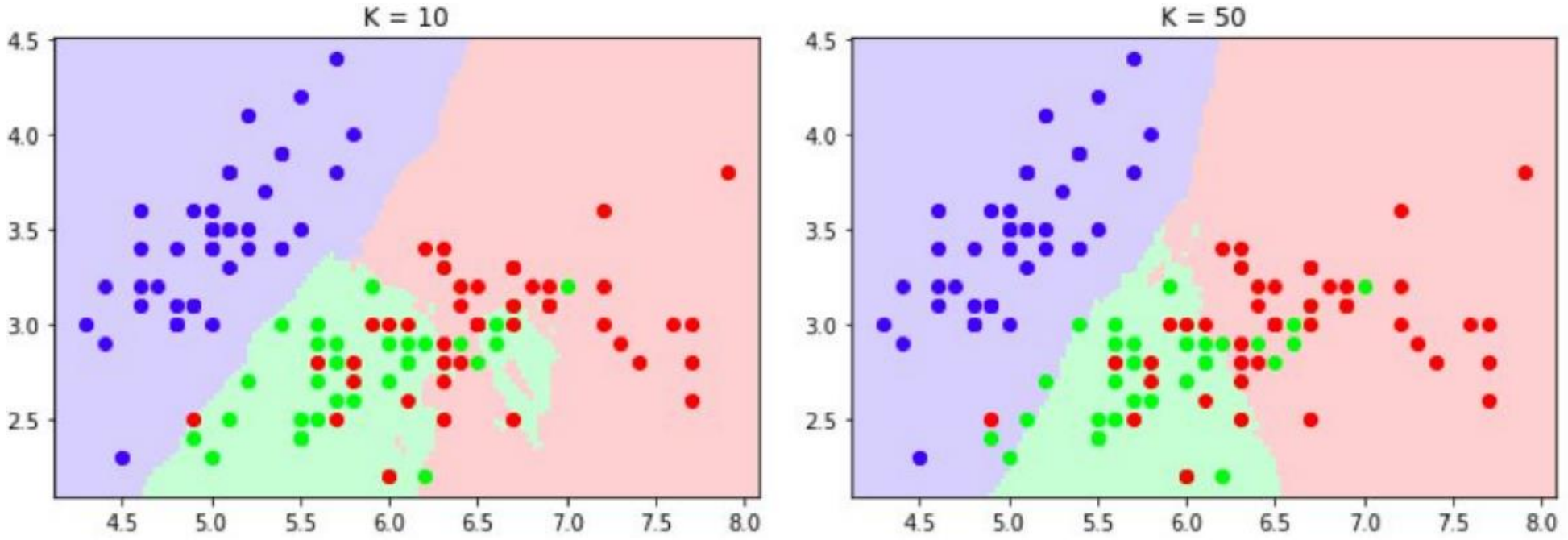
KNN DECISION BOUNDARY

- Piecewise linear decision boundary.
- Increasing k "simplifies" decision boundary
  - Majority voting means less emphasis on individual points

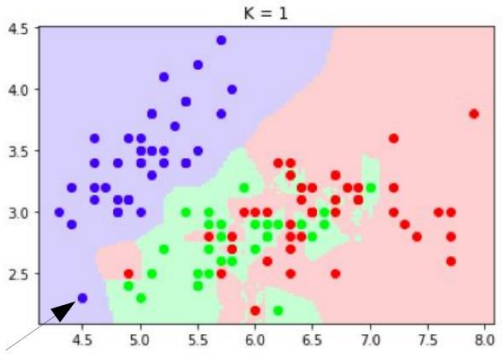


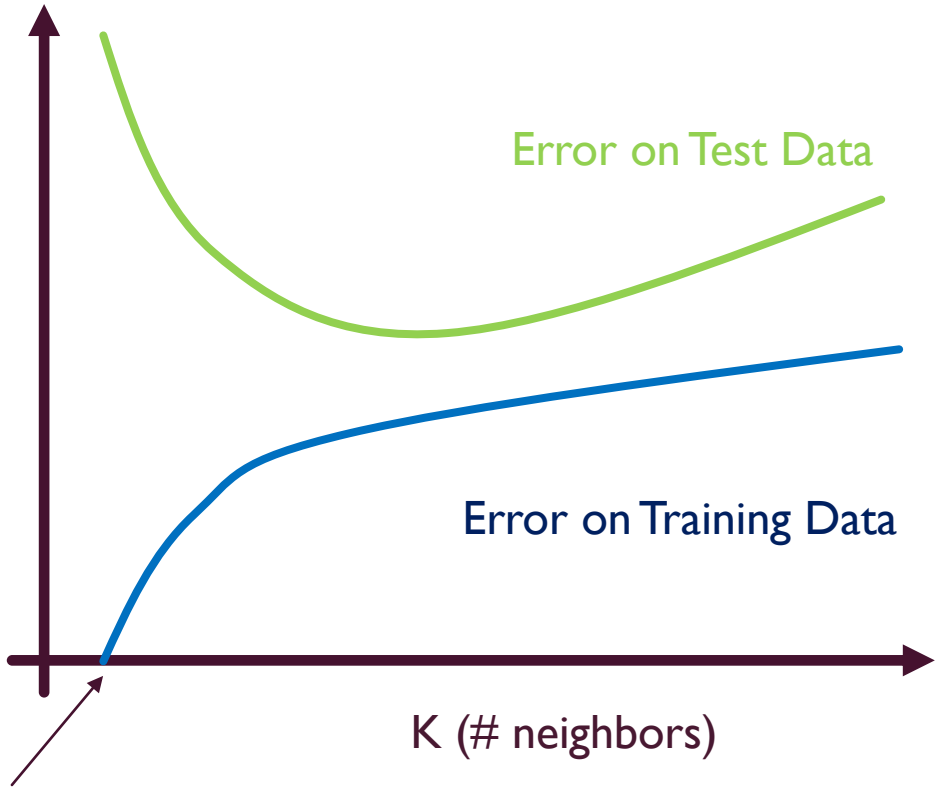
KNN DECISION BOUNDARY

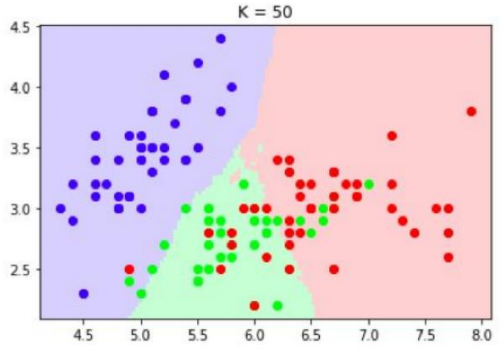
- Piecewise linear decision boundary.
- Increasing k "simplifies" decision boundary
  - Majority voting means less emphasis on individual points



ERROR RATES AND K







K=1? Zero error! Training data have been memorized

COMPLEXITY AND OVERFITTING

- Recap:
  - Complex model predicts all training points well and doesn't generalize to new data points.

- Theoretical considerations:
  - As k increases
    - We are averaging over more neighbors
    - The effective decision boundary is more "smooth"
    - As m (total number of training data) increases the optimal k value tends to increase (≈sqrt(n))

- Extensions of the Nearest neighbor classifier:
  - Weighted distances
    - E.g., some features may be more important others many be irrelevant
  - Fast search techniques (indexing) to find k-nearest points in d-space
  - Weighted average / voting based on distance

REFERENCE

- Multiclass classification: CS178: Machine Learning, Alexander Ihler, Xiaohui Xie, https://www.ics.uci.edu/~xhx/courses/CS273P/

https://towardsdatascience.com/animations-of-logistic-regression-with-python-31f8c9cb420