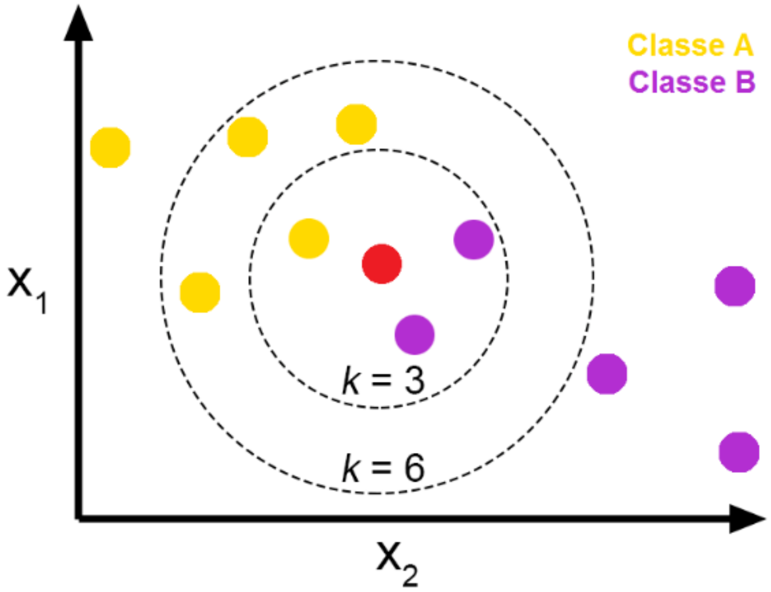
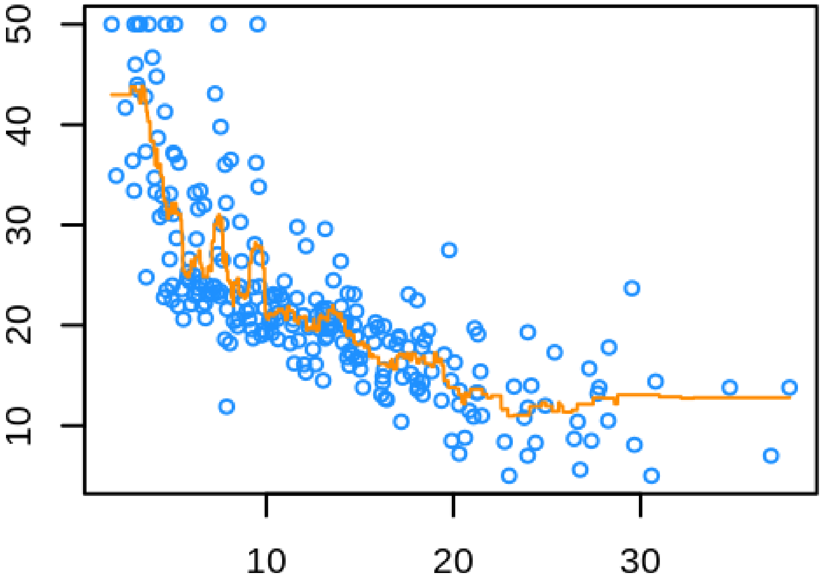
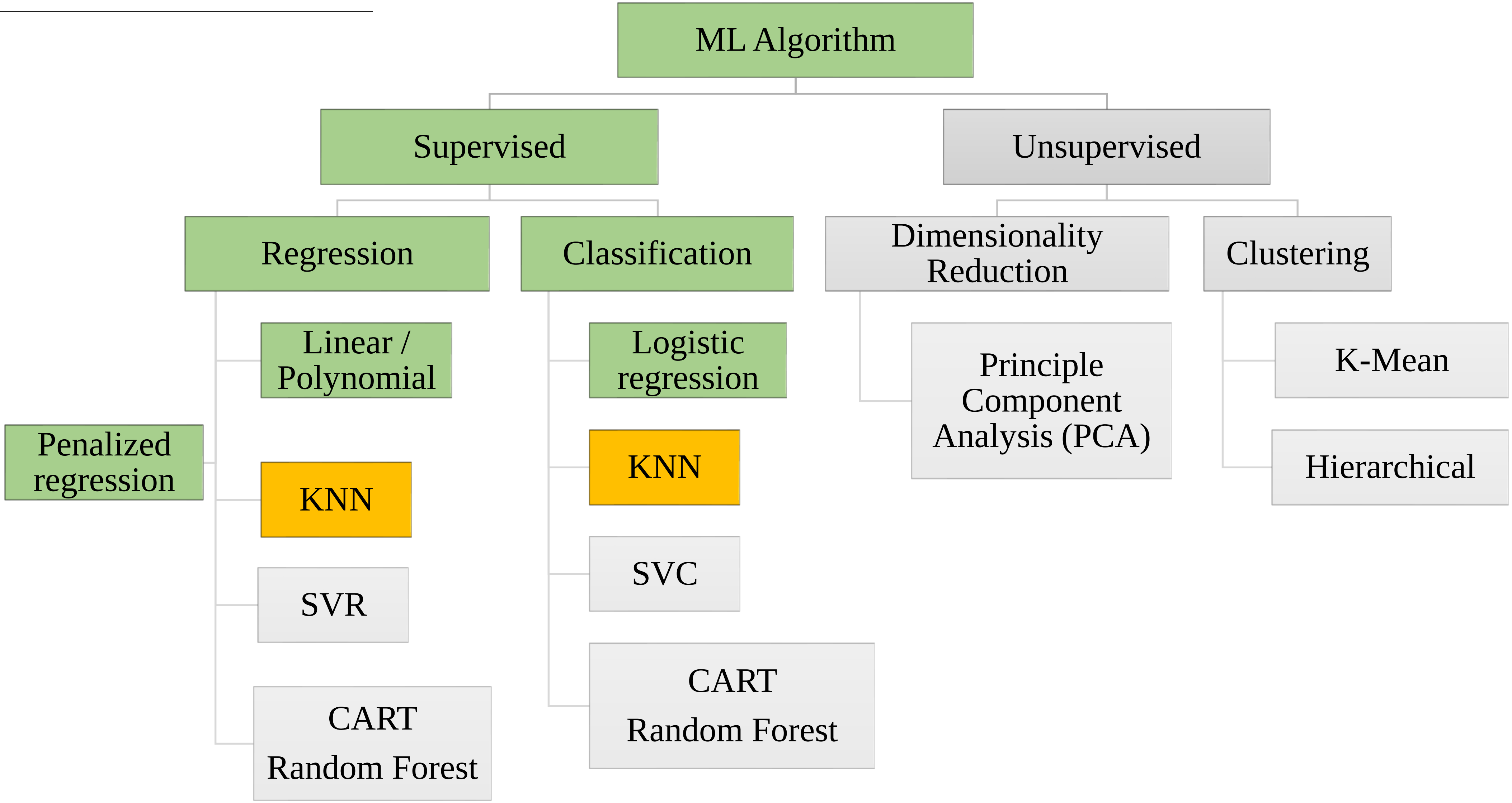
Class 12 – KNN
K-Nearest Neighbors

Prof. Pedram Jahangiry





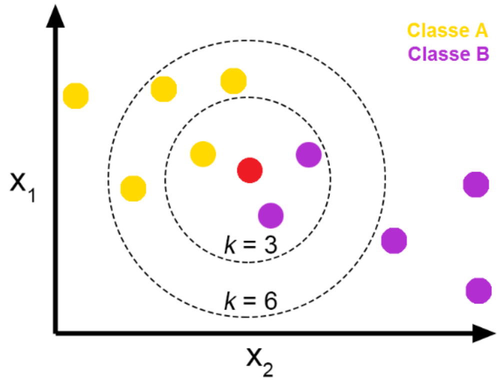
Road map


ML Algorithm
Supervised
Unsupervised
Regression
Classification
Dimensionality Reduction
Clustering
Linear / Polynomial
Penalized regression
KNN
SVR
CART
Random Forest
Logistic regression
KNN
SVC
CART
Random Forest
Principle Component Analysis (PCA)
K-Mean
Hierarchical

Topics

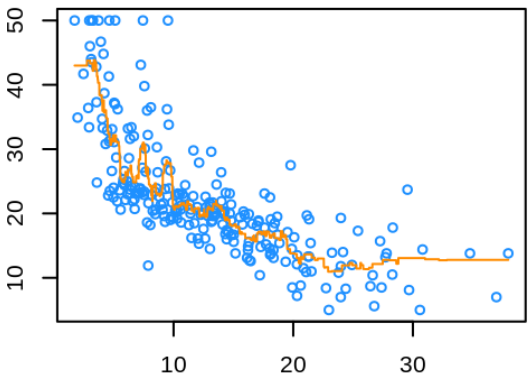
Part I

1. KNN Classification
2. Performance metrics and choice of K



Part II

1. KNN Regression
2. KNN vs Linear Regression
3. Performance metrics and choice of K



Part III

1. Curse of Dimensionality
2. Pros and Cons of KNN

Jon M. Huntsman School of Business
Utah State University

Part I

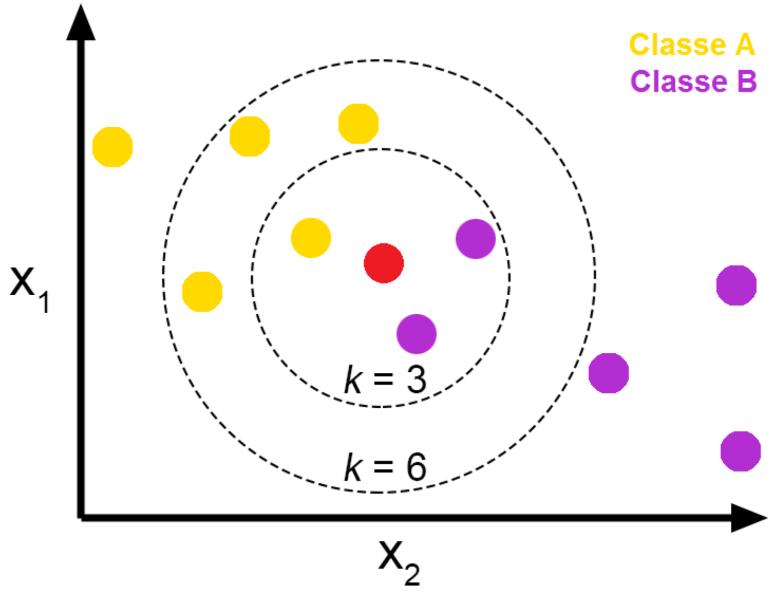
KNN Classification

KNN (K-Nearest Neighbors)

***K*-nearest neighbor (KNN)** is one of the simplest and best-known non-parametric **supervised** learning technique most often used for **classification**. The idea is to classify a new observation by finding similarities ("nearness") between it and its *k*-nearest neighbors in the existing dataset.

- Contrary to other learning algorithms that allow discarding the training data after the model is built, KNN keeps all training examples in memory.

- The choice of the **distance metric**, as well as the **value for *k***, are the choices the analyst makes before running the algorithm. So, these are hyperparameters.

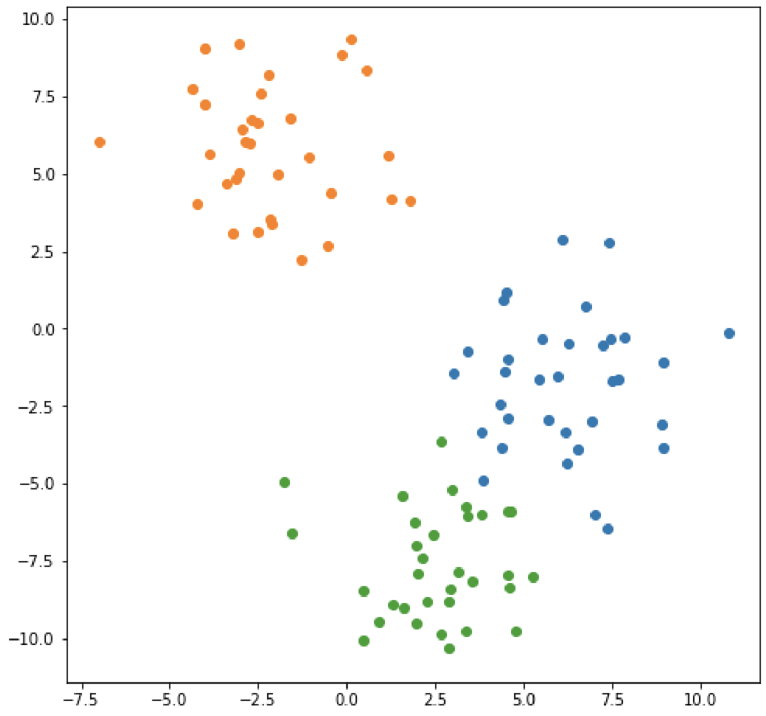


KNN steps

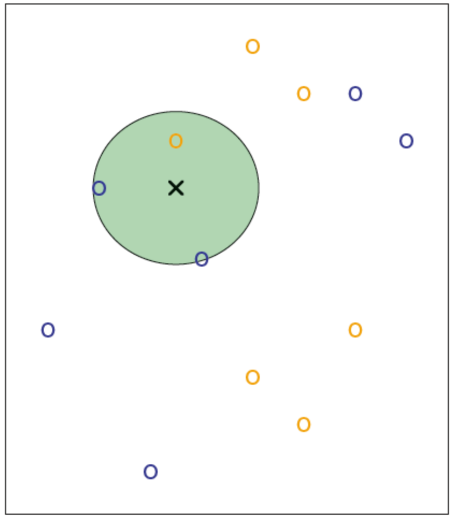
1. Choose number of neighbors K (positive integer)
2. Choose the distance metric (Minkowski, Euclidian, Manhattan, etc.)
3. Identify the K points in the training data that are closest to x_{te} in the test set. This neighborhood is represented by N_0.
4. Estimate the conditional probability for class j as the fraction of points in N_0 whose response values equal j:

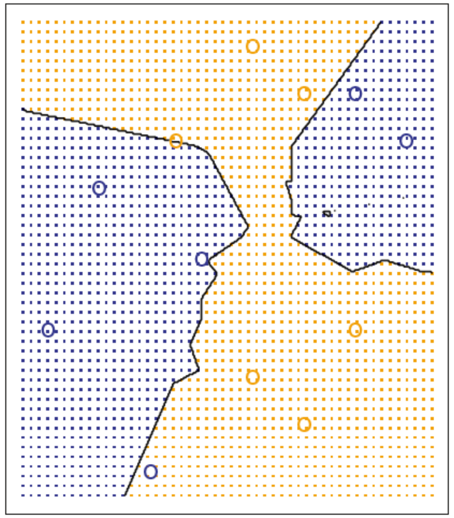
$$\Pr(Y = j \mid X = x_{te}) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j)$$

5. Classifies the test observation x_{te} to the class with the largest probability.



KNN Decision Boundary





- Two classes: blue and orange!
- Black cross: a test observation.
- When K=3, black cross is classified as:

  Blue
- Repeat this process for every potential element in the feature space!

- KNN decision boundary is shown in **black**.
- The blue grid indicates the region in which a test observation will be assigned to the blue class, and
- the orange grid indicates the region in which it will be assigned to the orange class.

Performance metrics

- Error rate = 1 – Accuracy

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP}$$

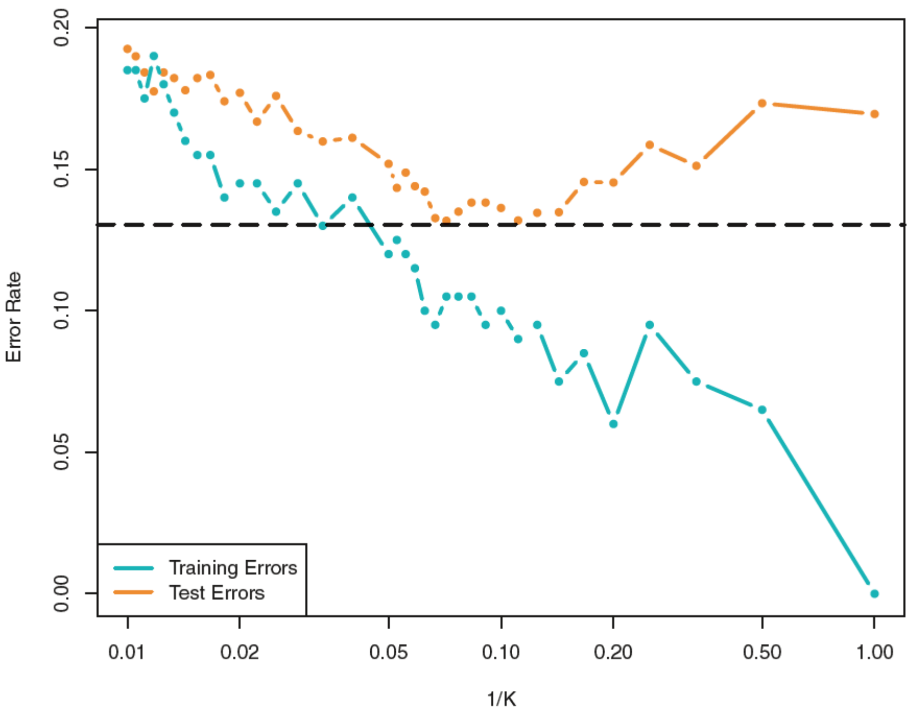
$$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$

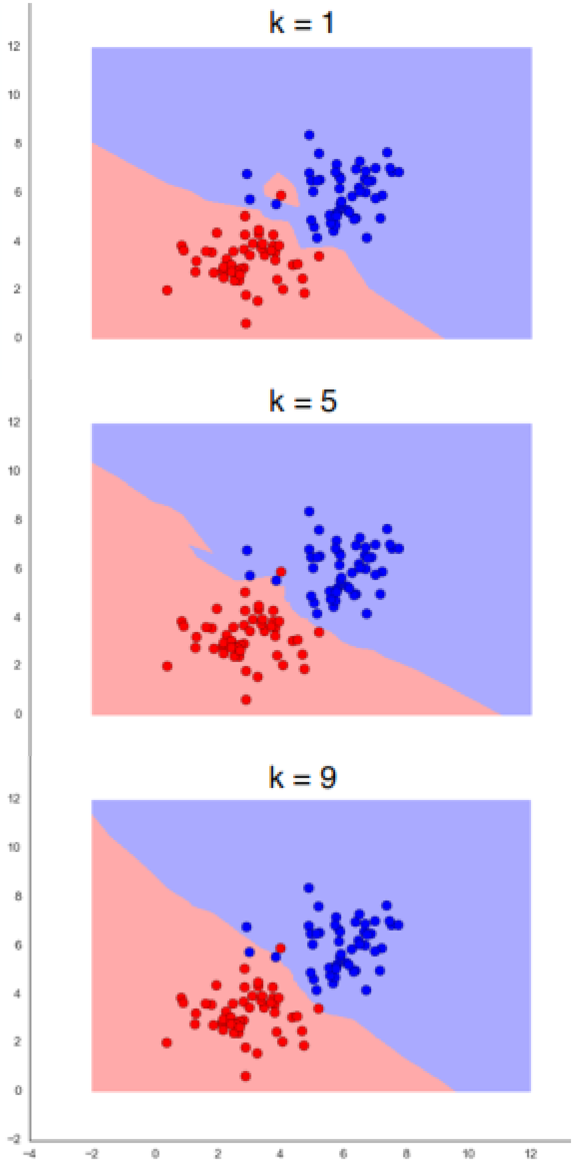
		Predictions	
		0 negative	1 positive
Actual	0 negative	TN	FP
	1 positive	FN	TP

- A good classifier is one for which the test error is smallest.
- Like any other classifier, if the data is highly imbalanced, then we should use f1score, precision and recall instead of the error rate.

Choice of K (Bias Variance Trade Off)

- K=1 very flexible model: Low bias but high variance.
- As K grows, less flexible model, decision boundary gets close to linear. This corresponds to a low variance but high bias.
- Optimal value of K:

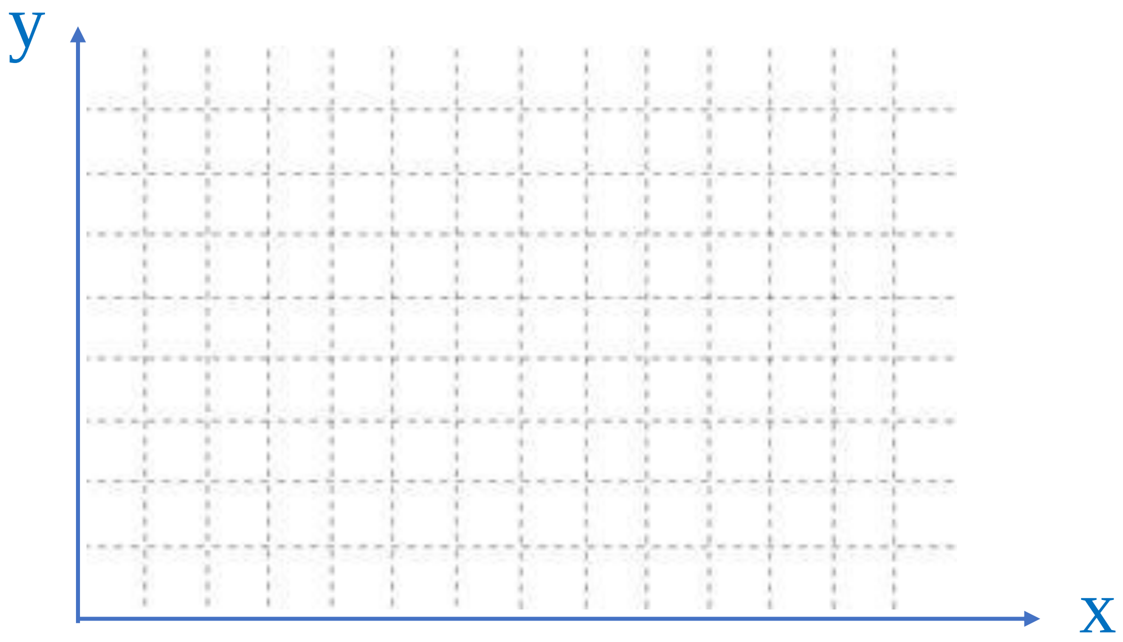




Part II
KNN Regression

KNN Regression

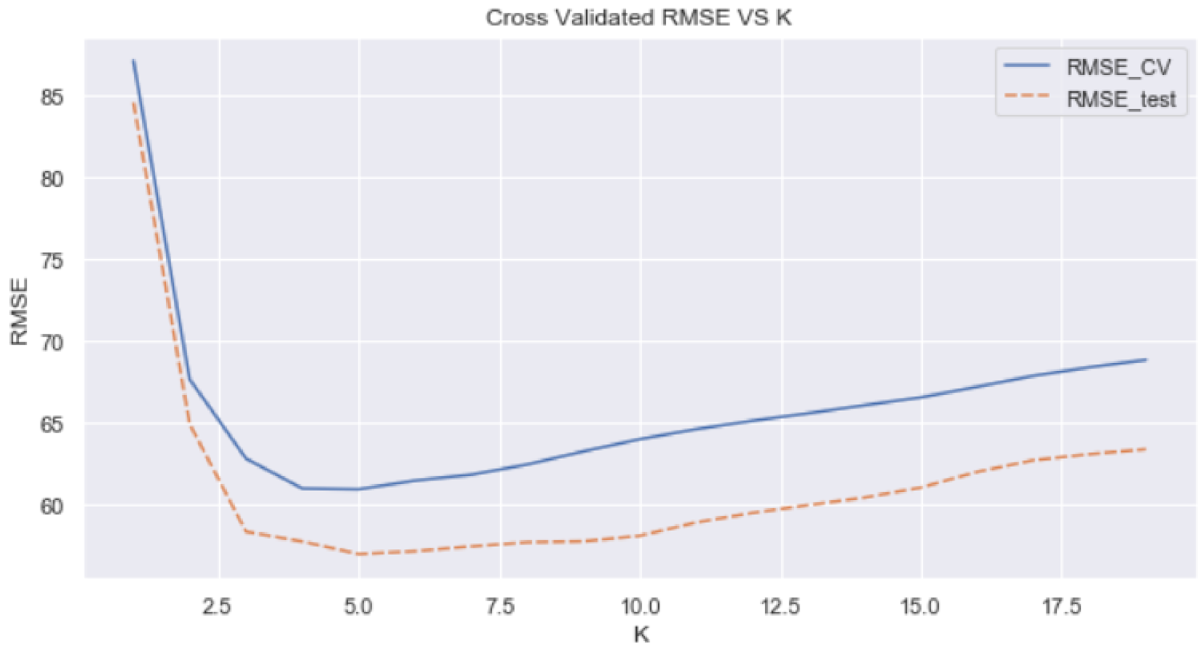
- The KNN regression method is closely related to the KNN classifier

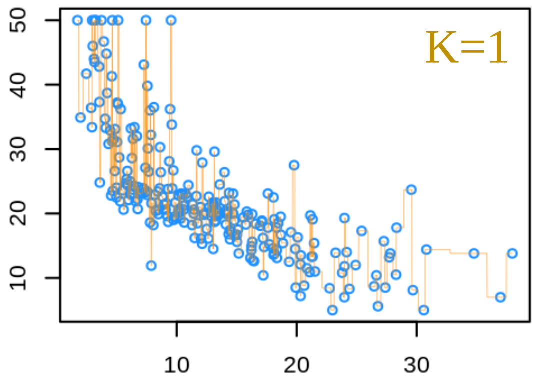


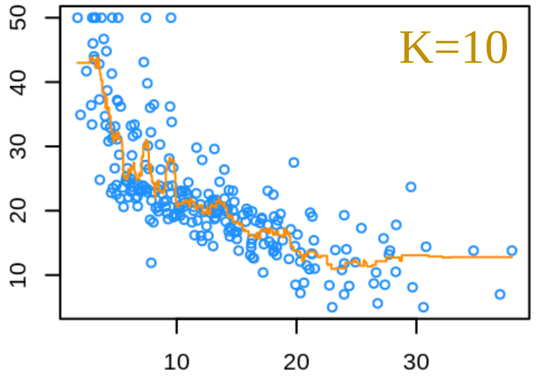
$$\hat{f}(x_{te}) = \frac{1}{K}\sum_{i \in N_0} y_i$$

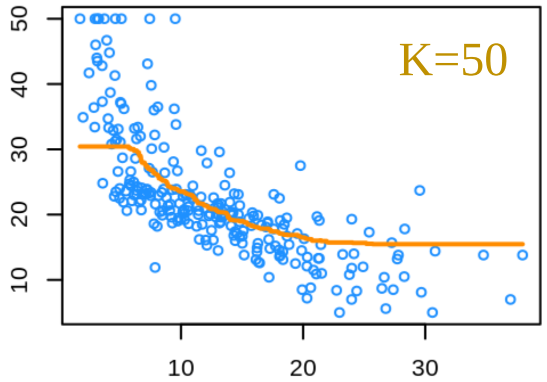
Choice of K (Bias Variance Trade Off)

- K=1 very flexible model: Low bias but high variance.
- As K grows, less flexible model, regression fit gets smoother and smoother. This corresponds to a low variance but high bias.
- Optimal value of K:



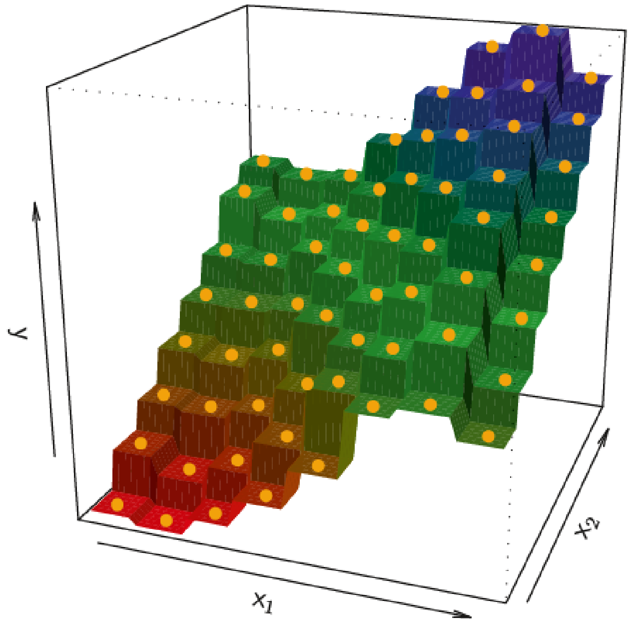


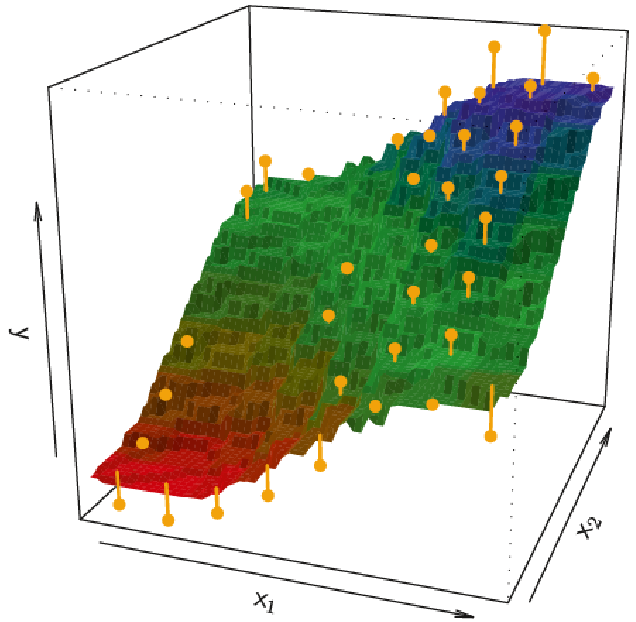




Choice of K (Bias Variance Trade Off)

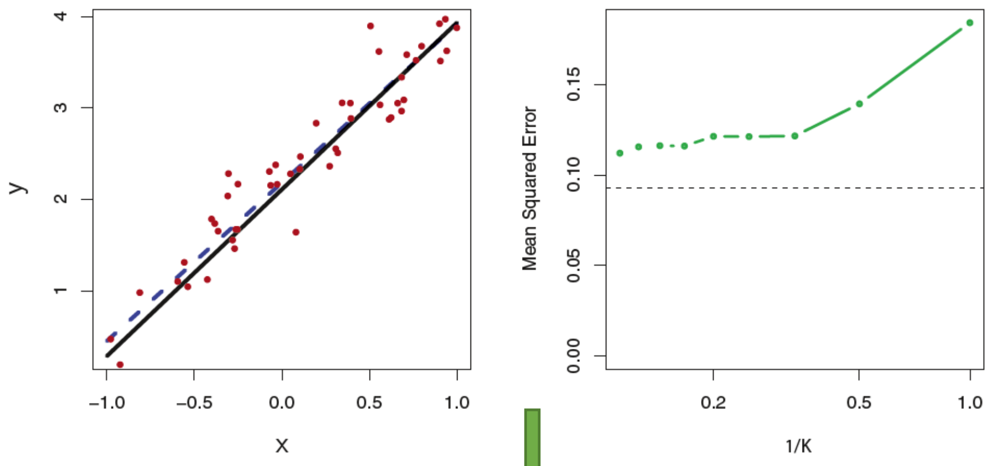
- K=1 very flexible model: Low bias but high variance.
- As K grows, less flexible model, regression fit gets smoother and smoother. This corresponds to a low variance but high bias.

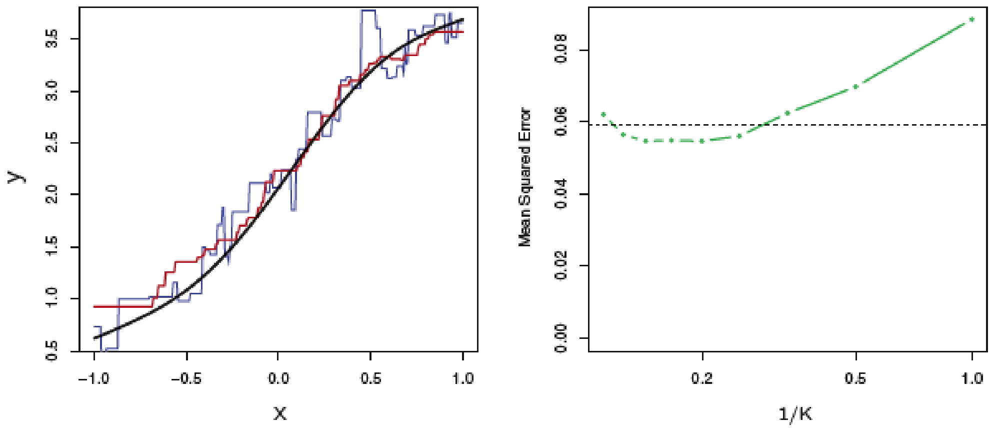


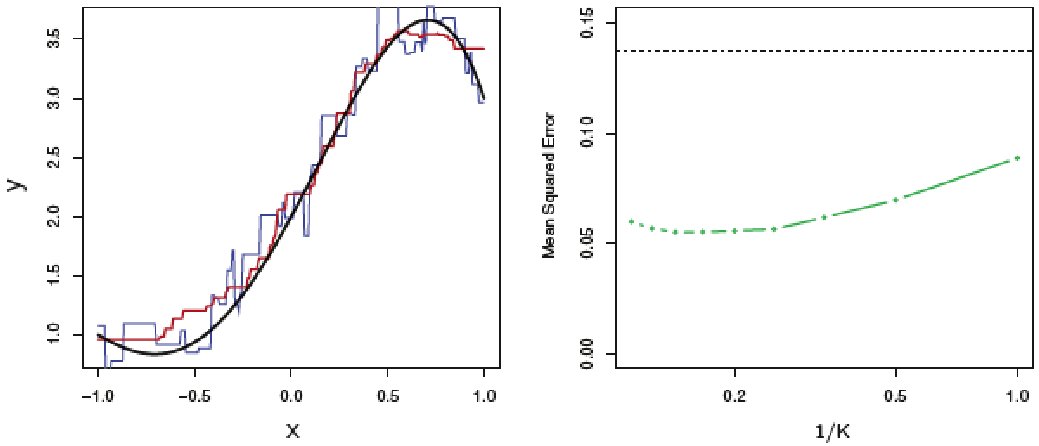


Linear regression vs KNN regression

- Black curve is the true relationship between y and X
- Green dashed line: KNN MSE_test
- Black dashed line: OLS MSE_test
- The more non-linear the true relationship, the better performance of KNN compared to OLS.



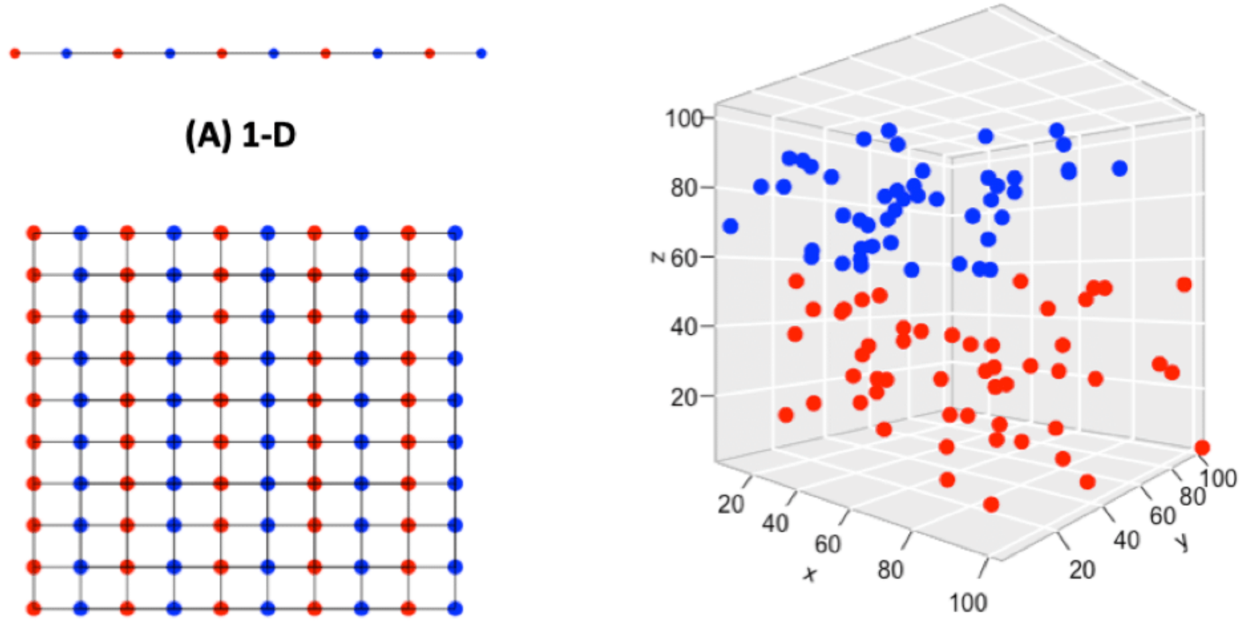




Part III
Pros and Cons

Curse of Dimensionality

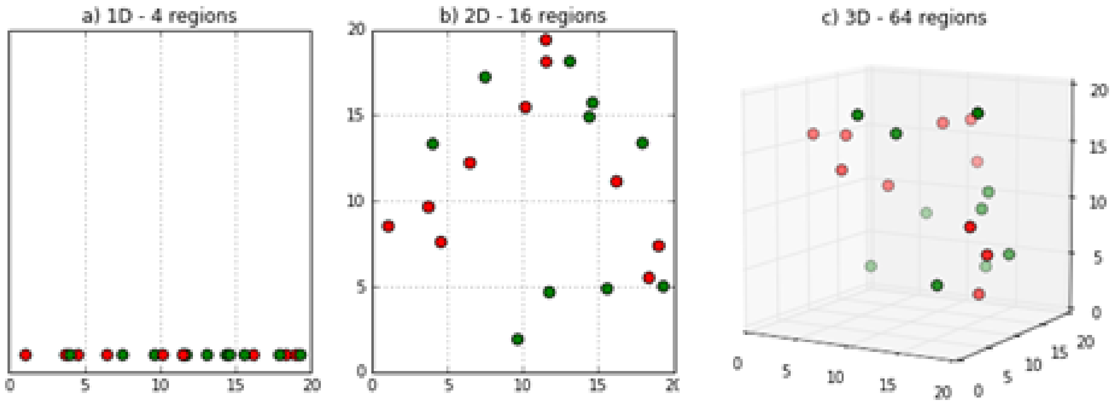
- The “Curse of Dimensionality” is a problem with the relationship between dimensionality and volume.
- Sparsity of data occurs when moving to higher dimensions. the volume of the space represented grows so quickly that the data cannot keep up and thus becomes sparse. (*Bellman, 1957*)

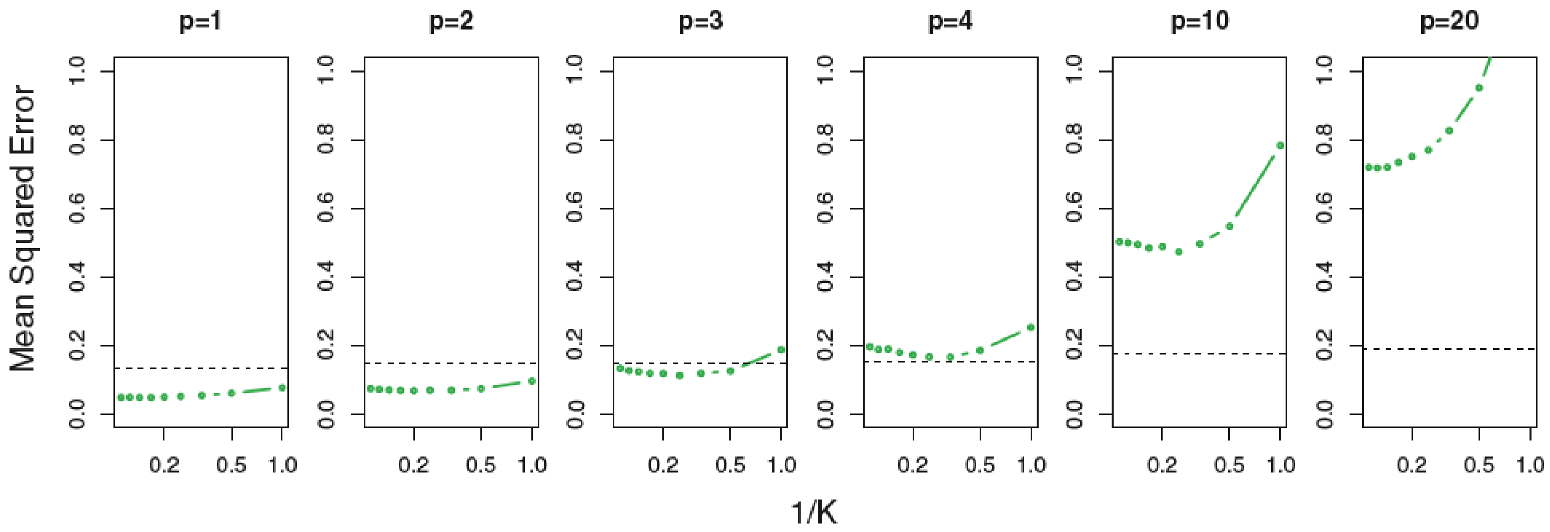


(B) 2-D

(C) 3-D

KNN and the Curse of Dimensionality





KNN’s Pros and Cons

Pros:

- Intuitive and simple
- No assumption (non-parametric)
- Easy to implement for multi-class problem
- Used both for classification and regression
- Few parameters/hyper-parameters

Cons:

- Slow (memory-based approach)
- Curse of dimensionality
- Not good with multiple categorical features
- Choice of K
- No interpretation (None!)

KNN Applications in finance

- Bankruptcy prediction
- Stock price prediction (buy/sell/hold)
- Corporate bond credit rating assignments
- Money laundering analysis
- Bank customer profiling
- Loan management
- Customized equity and bond index creation