Principles of Statistical Machine Learning
Introduction to Nearest Neighbors Learning

Fokoué Ernest, PhD
Professor of Statistics

@ErnestFokoue

To understand God's thoughts, one must study statistics, the measure of His purpose
Florence Nightingale

The Two Greatest Commandments

30 . . . ... and you shall love the Lord your God with all your heart and with all your soul and with all your mind and with all your strength.
31 The second is this: Love your neighbor as yourself. No other commandment is greater than these.

Mark 12:30

Learning Objectives

1. *Achieve a foundational knowledge of practical nearest neighbors classification learning*
2. *Discover and learn practical nearest neighbors regression learning*
3. *Understand the central role of similarity/proximity and dissimilarity measures in statistical machine learning*
4. *Dissect the most foundational properties of nearest neighbors learning machines*
5. *Explore and comprehend the strength and limitations of the Nearest Neighbors Learning paradigm*
6. *Apply nearest neighbors learning machines to practical problems*
7. *Read articles on applications of nearest neighbors learning*

Prerequisites and Tools

- *Prerequisites*
  1. *Understanding the basics of conditional probability*
  2. *Understanding indicator functions and summation notation*
  3. *Understanding of distances and metrics*
  4. *Basic knowledge of real analysis or advanced calculus*
  5. *Solid acquaintance with R and Rstudio from previous lectures*
- *Tools*
  1. `library(caret)`
  2. `library(MASS)`
  3. `library(class)`
  4. `library(FNN)`
  5. `knn(x,xnew,y,k,...)`

Who are my friends? Who are my neighbors?

When the character of a man is not clear to you, look at his friends

Japanese Proverb

Who is my neighbor? Who is my friend?

Out of all those we frequent

- *What qualifies some as our neighbors?*
- *What qualifies some as our friends?*
- *What does neighborhood have in common with friendship?*

Above all

- *How do we come up with a measurement or a measure that yield a number representing the magnitude of similarity, proximity, friendship, closeness, kindship?*
- *What do these philosophical and decidedly transcendental questions have to do with statistical machine learning?*

Answer.

- *Similarity, Proximity and Neighborhood are the central tenets and building blocks of the nearest neighbors learning paradigm*

Motivating Binary Classification Example in 2D

To gain insights into the nearest neighbors learning paradigm, let's consider the following simple Binary Classification Example in 2D.

	x_1	x_2	y
Peter	*1.50*	*3.00*	*1*
Andrew	*2.00*	*2.00*	*1*
James	*2.50*	*4.00*	*1*
Paul	*4.00*	*3.00*	*1*
Steven	*3.00*	*3.00*	*1*
Matthew	*6.00*	*5.50*	*0*
Mark	*7.00*	*6.00*	*0*
Luke	*7.00*	*5.00*	*0*
John	*8.50*	*6.00*	*0*

Table: *Set of people with known labels.*

$\mathscr{D}_n = \{(\mathbf{x}_i, \mathrm{y}_i) \overset{iid}{\sim} p_{\mathrm{xy}}(\mathbf{x}, \mathrm{y}),\ \mathbf{x}_i \in \mathscr{X}, \mathrm{y}_i \in \mathscr{Y},\ i = 1, \cdots, n\}$, *where in this case* $\mathscr{X} \subseteq \mathbb{R}^2$ *and* $\mathscr{Y} = \{0, 1\}$.

Motivating Binary Classification Example in 2D

To help gain insights into the principles of kNearest Neighbors classification, we consider the deliberately very simple and straightforward 2D binary classification task shown below

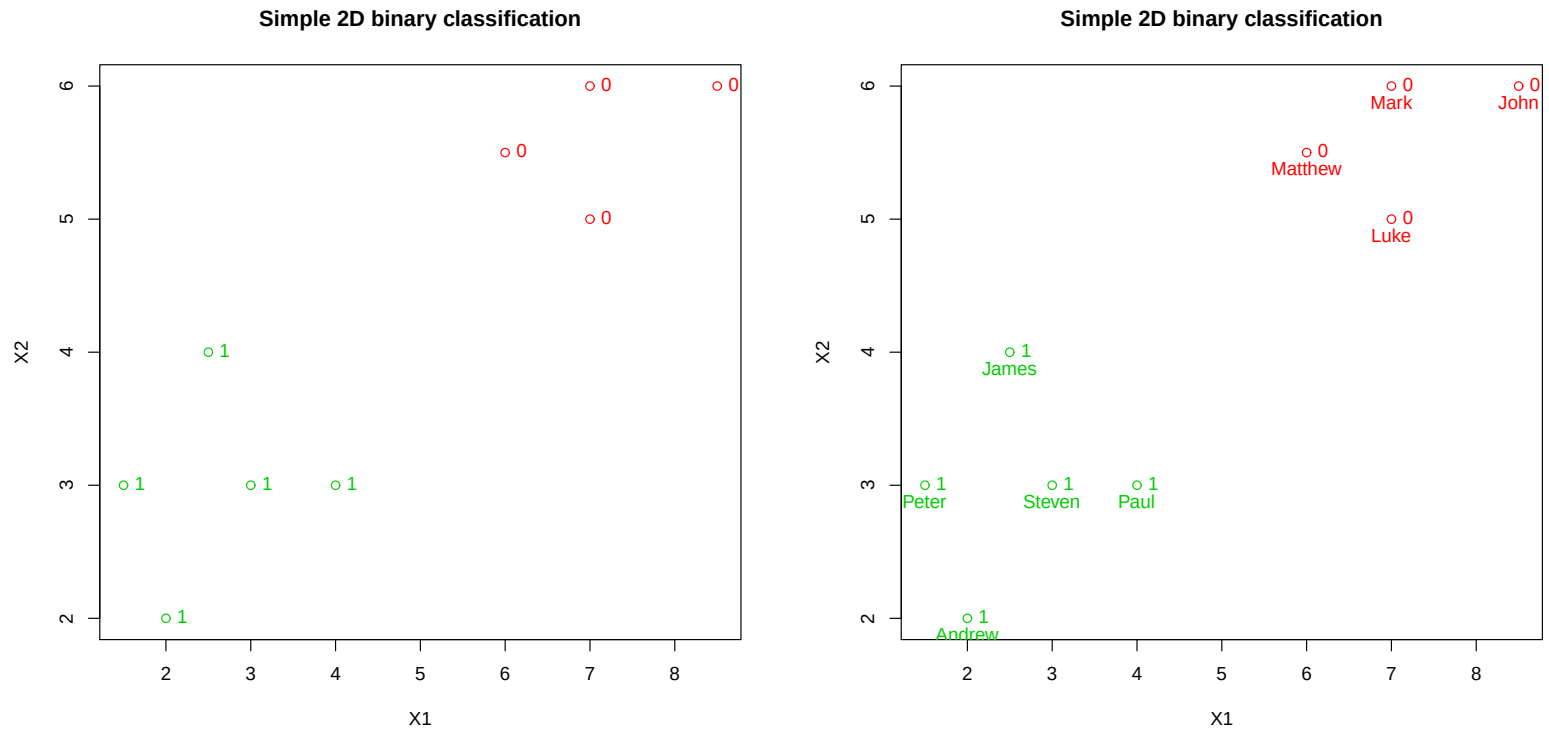


Figure: (left) Plot without names . (right) Plot with names.

Motivating Binary Classification Example in 2D

Given $\mathscr{D}_n = \{(\mathbf{x}_i, \mathrm{y}_i) \overset{iid}{\sim} p_{\mathrm{xy}}(\mathbf{x}, \mathrm{y}),\ \mathbf{x}_i \in \mathscr{X}, \mathrm{y}_i \in \mathscr{Y},\ i = 1, \cdots, n\}$, *encountered earlier with people whose character* Y *is known, What is the character of these new people given in the following table* (2)

	x_1	x_2	y
Timothy	*6.00*	*4.50*	*?*
Barnabas	*5.00*	*4.00*	*?*
Silas	*3.50*	*4.00*	*?*

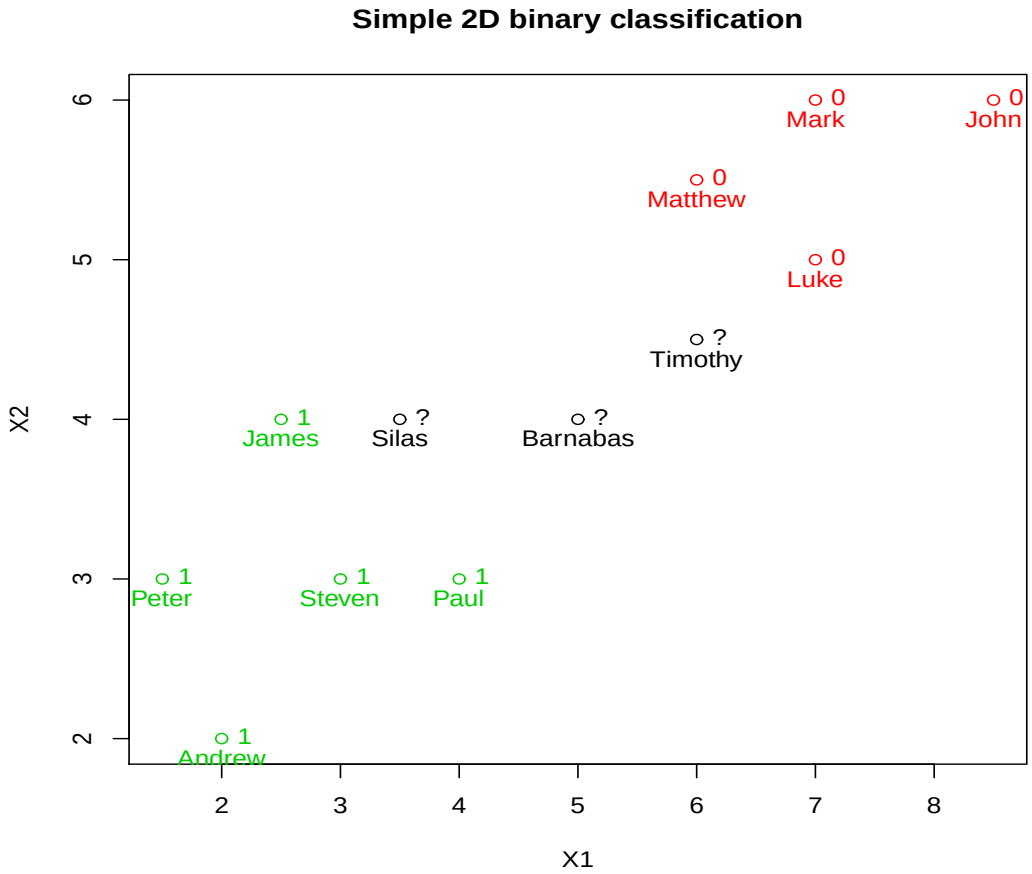
Table: *Set of new people who labels need to be predicted.*

Questions for investigation

- *Out of all the existing people with known characteristics like Peter, Andrew, James, Paul, Steven, Matthew, Mark , Luke, John, who are the people Timothy is closest or more similar to?*
- *Knowing the labels of Peter, Andrew, James, Paul, Steven, Matthew, Mark , Luke, John, what then is the most plausible label for Timothy?*

Motivating Binary Classification Example in 2D

Task: *We are given new people* {*Timothy, Barnabas, Silas*} *as shown below, and we want to assign each one of them to their "correct" class.*



We will herein build the so-called k-Nearest Neighbors (kNN) classifier.

Methodological insights

1. *Similarity:* *What sort of mathematical object* $d : \mathscr{X} \times \mathscr{X} \longrightarrow \mathbb{R}_+$ *exist or can be constructed such that*

$$d(\mathbf{x}_{\texttt{Timothy}}, \mathbf{x}_i) = \mathbf{similarity}(\texttt{Timothy}, \texttt{Person}_i), \quad i \in [n]$$

may be used to find the nearest neighbors of Timothy in $\mathscr{D}_n$*?*

2. *Complexity:* *How many neighbors (proxies) are ultimately needed to full grasped the desired prediction?*
3. *Estimation and prediction:* *How to construct a learning machine* $f : \mathscr{X} \longrightarrow \mathscr{Y}$ *such that*

$$\mathbf{class}(\texttt{Timothy}) = f(\mathbf{x}_{\texttt{Timothy}}) \in \mathscr{Y} = \{0, 1\} \tag{1}$$

4. *How good is* f*?:* *How often does* f *misclassify? How good is* f*?*

$$\Pr[Y_{\texttt{Timothy}} \neq f(\mathbf{x}_{\texttt{Timothy}})]$$

Philosophical Basis of the Nearest Neighbors Paradigm

Tell me who your friends are, and I will tell you who you are.
Mexican Proverb

Intuition of kNearest Neighbors Classification

- *k-Nearest Neighbors Principle:* *The reasonable class/category for a given object is the most prevalent class among its nearest neighbors*

- *k-Nearest Neighbors Steps:* *Given a new point to be classified,*
  - *Choose a distance for measuring how far a given point is from another*
  - *Set the size of the neighborhood k*
  - *Compute the distance from each existing point to the new point*
  - *Identify the class labels of the k points closest/nearest to the new point*
  - *Assign the most frequent label to the new point*

- *k-Nearest Neighbors Classification:* *The estimated class of a vector $\mathbf{x}$ is the most frequent class label in the neighborhood of $\mathbf{x}$.*

Objects, Vectors, Data Matrix, and Similarity Measures

- *Let* $X = (X_1, X_2, \cdots, X_p)^\top$ *where each* X_j *is one of* p *variables*
- *For the most common type of tasks,* $X_l \in \mathbb{R}$ *is a real number*
- *For some more advanced tasks, like text mining, we'll consider cases where* X_l *can be of any type.*
- *Given* $\mathscr{D}_n$ *with* $\mathbf{x}_i = (\mathrm{x}_{i1}, \cdots, \mathrm{x}_{ip})^\top \in \mathscr{X}$*, the data matrix is*

$$\boldsymbol{X} = \begin{bmatrix} \mathrm{x}_{11} & \mathrm{x}_{12} & \cdots & \mathrm{x}_{1p} \\ \mathrm{x}_{21} & \mathrm{x}_{22} & \cdots & \mathrm{x}_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{x}_{n1} & \mathrm{x}_{n2} & \cdots & \mathrm{x}_{np} \end{bmatrix}$$

- *Goal: We are interested in the distances b/w the rows of* $\boldsymbol{X}$*, i.e.*

$$d(\mathbf{x}_i, \mathbf{x}_j) \quad \text{for} \quad i > j$$

Basic Properties of a Distance (Metric)

A distance on an input space $\mathscr{X}$ is a bivariate function $d(\cdot, \cdot)$, that is

$$d : \mathscr{X} \times \mathscr{X} \longrightarrow [0, +\infty)$$

with the following properties

1. $d(\mathbf{x}, \mathbf{y}) \geq 0$ *(Nonnegativity)*
2. $d(\mathbf{x}, \mathbf{y}) = 0$ *if and only if* $\mathbf{x} = \mathbf{y}$ *(identity of indiscernibles)*
3. $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ *(Symmetry)*
4. $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})$ *(Subadditivity or triangle inequality)*

Some examples of metrics (distances) include

- $d(\mathrm{x}, \mathrm{y}) = |\mathrm{x} - \mathrm{y}|$ *(univariate ℓ_1 distance)*
- $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_1 = \sum_{j=1}^{p} |\mathrm{x}_j - \mathrm{y}_j|$ *(multivariate ℓ_1 distance)*

Distances as Similarity Measures

Given two vectors $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^p$, The most common distances are

- *Euclidean distance: ℓ_2 distance*

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{\ell=1}^{p} (\mathrm{x}_{i\ell} - \mathrm{x}_{j\ell})^2} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$$

- *Manhattan distance: city block or ℓ_1 distance*

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{\ell=1}^{p} |\mathrm{x}_{i\ell} - \mathrm{x}_{j\ell}| = \|\mathbf{x}_i - \mathbf{x}_j\|_1$$

- *Maximum distance: infinity or Supremum distance*

$$d(\mathbf{x}_i, \mathbf{x}_j) = \max_{\ell=1,\cdots,p} |\mathrm{x}_{i\ell} - \mathrm{x}_{j\ell}| = \|\mathbf{x}_i - \mathbf{x}_j\|_\infty$$

Other common distances: (a) Minkowski*; (b)* canberra*; (c)* binary*.*

Distances as Similarity Measures

Given two vectors $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^p$, The most common distances are

- *Minkowski distance: ℓ_q distance*

$$d(\mathbf{x}_i, \mathbf{x}_j) = \left\{ \sum_{\ell=1}^{p} |\mathrm{x}_{i\ell} - \mathrm{x}_{j\ell}|^q \right\}^{1/q}$$

- *Canberra distance:*

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{\ell=1}^{p} \frac{|\mathrm{x}_{i\ell} - \mathrm{x}_{j\ell}|}{|\mathrm{x}_{i\ell} + \mathrm{x}_{j\ell}|}$$

- *Jaccard/Tanimoto distance: For binary vectors ie $\mathbf{x}_i \in \{0,1\}^p$*

$$d(\mathbf{x}_i, \mathbf{x}_j) = 1 - \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{|\mathbf{x}_i|^2 + |\mathbf{x}_j|^2 - \mathbf{x}_i \cdot \mathbf{x}_j}$$

$\mathbf{x}_i \cdot \mathbf{x}_j = \sum_{\ell=1}^{p} \mathrm{x}_{i\ell}\mathrm{x}_{j\ell} = \sum_{\ell=1}^{p} \mathrm{x}_{i\ell} \wedge \mathrm{x}_{j\ell}$ *and* $|\mathbf{x}_i|^2 = \sum_{\ell=1}^{p} \mathrm{x}_{i\ell}^2$

Intuition of kNearest Neighbor Classification

We consider applying the kNN algorithm to the above simple example. First, let's use kNN to find the neighbors of Timothy = (6.0, 4.5). Matthew is his nearest neighbor. Luke is his second nearest neighbor. Mark is his third nearest neighbor.

	X1	X2	class	dist	rank
Peter	1.5	3.0	1	22.50	9
Andrew	2.0	2.0	1	22.25	8
James	2.5	4.0	1	12.50	7
Paul	4.0	3.0	1	6.25	4
Steven	3.0	3.0	1	11.25	6
Matthew	6.0	5.5	0	1.00	1
Mark	7.0	6.0	0	3.25	3
Luke	7.0	5.0	0	1.25	2
John	8.5	6.0	0	8.50	5

Question: What is the correct class for Timothy?

Intuition of kNearest Neighbor Classification

What are the 3 nearest neighbors of Barnabas = (5.0, 4.0)?

	X1	X2	class	dist	rank
Peter	1.5	3.0	1	13.25	8
Andrew	2.0	2.0	1	13.00	7
James	2.5	4.0	1	6.25	5
Paul	4.0	3.0	1	2.00	1
Steven	3.0	3.0	1	5.00	3
Matthew	6.0	5.5	0	3.25	2
Mark	7.0	6.0	0	8.00	6
Luke	7.0	5.0	0	5.00	4
John	8.5	6.0	0	16.25	9

Question: What is the correct class for Barnabas?

Intuition of kNearest Neighbor Classification

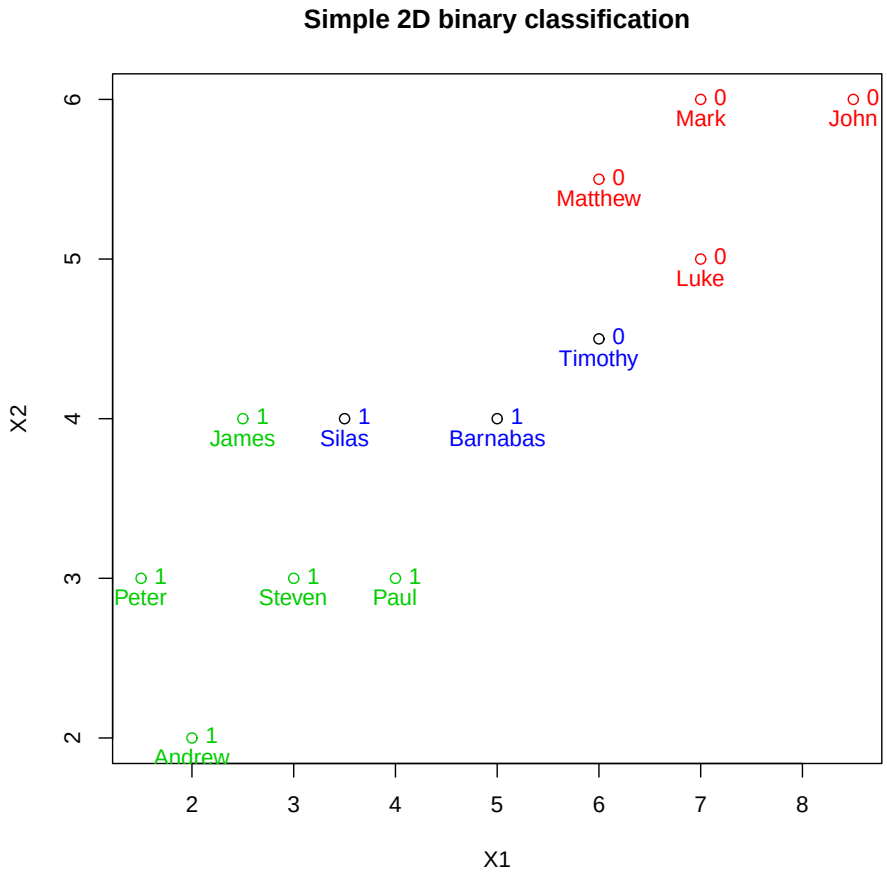
Finally, determine the 4 *nearest neighbors of Silas* $=$ *(3.5, 4.0)*

	X1	X2	class	dist	rank
Peter	1.5	3.0	1	5.00	4
Andrew	2.0	2.0	1	6.25	5
James	2.5	4.0	1	1.00	1
Paul	4.0	3.0	1	1.25	2
Steven	3.0	3.0	1	1.25	3
Matthew	6.0	5.5	0	8.50	6
Mark	7.0	6.0	0	16.25	8
Luke	7.0	5.0	0	13.25	7
John	8.5	6.0	0	29.00	9

Question: What is the correct class for Silas?

Intuition of kNearest Neighbor Classification

The final classification of all the three new individuals is below



We now give more ample description of the kNearest Neighbors Algorithm.

kNearest Neighbors (kNN) classification

$$\mathscr{D}_n = \{(\mathbf{x}_i, \mathrm{y}_i) \overset{iid}{\sim} p_{\mathrm{xy}}(\mathbf{x}, \mathrm{y}),\ \mathbf{x}_i \in \mathscr{X}, \mathrm{y}_i \in \mathscr{Y},\ i \in [n]\},\ \mathscr{Y} = \{1, \cdots, G\}.$$

1. *Choose the value of k and the distance to be used*
2. *Let $\mathbf{x}$ be a new point. Compute*

$$d_i = d(\mathbf{x}, \mathbf{x}_i) \qquad i = 1, \cdots, n$$

3. *Rank all the distances d_i in increasing order*

$$d_{(1)} \leq d_{(2)} \leq \cdots \leq d_{(k)} \leq d_{(k+1)} \leq \cdots \leq d_{(n)}$$

4. *Form $\mathcal{V}_k(\mathbf{x})$, the k-Neighborhood of $\mathbf{x}$*

$$\mathcal{V}_k(\mathbf{x}) = \left\{\mathbf{x}_i :\ d(\mathbf{x}, \mathbf{x}_i) \leq d_{(k)}\right\}$$

5. *Compute the predicted response $\widehat{Y}$ as*

$$\widehat{Y}_{\texttt{kNN}} = \widehat{f}_{\texttt{kNN}}(\mathbf{x}) = \textit{Most frequent label in } \mathcal{V}_k(\mathbf{x})$$

k-Nearest Neighbors (kNN) classification

1. *Predicted Response:* Compute the predicted response $\widehat{Y}$ as

$$\widehat{Y}_{\texttt{kNN}} = \widehat{f}_{\texttt{kNN}}(\mathbf{x}) = \underset{g \in \{1, \cdots, G\}}{\texttt{argmax}} \left\{ p_g^{(k)}(\mathbf{x}) \right\}$$

where

$$p_g^{(k)}(\mathbf{x}) = \frac{1}{k} \sum_{i=1}^{n} \mathbb{1}(\mathbf{x}_i \in \mathcal{V}_k(\mathbf{x})) \mathbb{1}(Y_i = g)$$

estimates the probability that $\mathbf{x}$ *belongs to class* g *based on* $\mathcal{V}_k(\mathbf{x})$.

2. *Posterior probability estimate:* Indeed, $p_g^{(k)}(\mathbf{x})$ *can be thought of as a rough estimate of* $\pi_g(\mathbf{x}) = \Pr[Y = g | \mathbf{x}]$, *the posterior probability of class membership of* $\mathbf{x}$, *ie*

$$p_g^{(k)}(\mathbf{x}) = \frac{1}{k} \sum_{i=1}^{n} \mathbb{1}(\mathbf{x}_i \in \mathcal{V}_k(\mathbf{x})) \mathbb{1}(Y_i = g) \approx \widehat{\pi_j(\mathbf{x})}$$

Basic Remarks on Classification

- *Finding an automatic classification rule that achieves the absolute very best on the present data is not enough since infinitely many more observations can be generated by $\Psi(\mathbf{x}, y)$ for which good classification will be required.*
- *Even the universally best classifier will make mistakes.*
- *Of all the functions in $\mathscr{Y}^{\mathscr{X}}$, it is reasonable to assume that there is a function $f^\star$ that maps any $\mathbf{x} \in \mathscr{X}$ to its corresponding $y \in \mathscr{Y}$, i.e.,*

$$\begin{array}{rccc} f^\star: & \mathscr{X} & \to & \mathscr{Y} \\ & \mathbf{x} & \mapsto & f^\star(\mathbf{x}), \end{array}$$

with the minimum number of mistakes.

Loss and Risk in Pattern Recognition

For this classification/pattern recognition, the so-called 0-1 *loss function defined below is used. More specifically,*

$$\ell(y, f(\mathbf{x})) = \mathbb{1}\{y \neq f(\mathbf{x})\} = \begin{cases} 0 & \text{if} \quad y = f(\mathbf{x}), \\ 1 & \text{if} \quad y \neq f(\mathbf{x}). \end{cases} \tag{2}$$

The corresponding risk functional is

$$R(f) = \int \ell(y, f(\mathbf{x})) d\Psi(\mathbf{x}, y) = \mathbb{E}\left[\mathbb{1}\{Y \neq f(X)\}\right] = \underset{(X,Y)\sim\Psi}{\Pr}[Y \neq f(X)].$$

The minimizer of the 0-1 *risk functional over all possible classifiers is the so-called* Bayes classifier *which we shall denote here by* $f^\star$ *given by*

$$f^\star = \arg\inf_f \left\{ \underset{(X,Y)\sim\Psi}{\Pr}[Y \neq f(X)] \right\}.$$

Specifically, the Bayes' classifier $f^\star$*, whose risk is* $R^\star = R(f^\star)$*, is given by the posterior probability of class membership, namely*

$$f^\star(\mathbf{x}) = \arg\max_{y\in\mathscr{Y}} \left\{ \Pr[Y = y|\mathbf{x}] \right\}.$$

kNearest Neighbors (kNN) classification

Some properties of kNN estimators include

- *kNearest Neighbors (kNN) essentially performs classification by voting for the most popular response among the k nearest neighbors of $\mathbf{x}$.*
- `kNN` *provides the most basic form of nonparametric classification*
- *Thanks to the fact that the estimated response $\widehat{Y}_{\texttt{kNN}}$ for $\mathbf{x}$ is - at least - a crude nonparametric estimator of Bayes classifier's response*
- *Since the fundamental building block of* `kNN` *is the distance measure, one can easily perform classification beyond the traditional setting where the predictors are numeric. For instance, classification with kNN can be readily performed on indicator attributes*

$$\mathbf{x}_i = (\mathrm{x}_{i1}, \cdots, \mathrm{x}_{ip})^\top \in \{0, 1\}^p$$

- `kNN` *classifiers are inherently naturally multi-class, and are used extensively in applications such as image processing, character recognition and general pattern recognition tasks*

kNearest Neighbors (kNN) classification

Limitations of the basic kNearest Neighbors approach:

1. *Equidistance: All neighbors are given the same contribution to the estimate of the response; Indeed, in the estimated probability*

$$p_g^{(k)}(\mathbf{x}) = \sum_{i=1}^{n} w_i(\mathbf{x}) \mathbb{1}(\mathbf{x}_i \in \mathcal{V}_k(\mathbf{x})) \mathbb{1}(\mathrm{y}_i = g)$$

   the weight $w_i(\mathbf{x}) = \frac{1}{k} = \texttt{constant}$ *for all points in* $\mathcal{V}_k(\mathbf{x})$ *regardlessly of how far they are from* $\mathbf{x}$.

2. *No model, no interpretability: There is no underlying model, therefore no interpretation of the response relative to the predictor variables. There is no training per se, since all happens at prediction. For this reason,* `kNN` *is referred to as lazy method.*

3. *Computationally intensive: Predictions are computationally very intensive, due to the fact that for each new observation, the whole dataset must be traversed to compute the response*

Weighted kNearest Neighbors Classification

kNN classification can be improved by weighting the votes as a function of the distance from $\mathbf{x}$*. Some of the common weighting schemes include:*

- *Exponential decay:*

$$w_i = \frac{e^{-d_i}}{\sum_{l=1}^{k} e^{-d_l}}$$

- *Inverse distance:*

$$w_i = \frac{\frac{1}{1+d_i}}{\sum_{l=1}^{k} \frac{1}{1+d_l}}$$

The weights are defined so as to preserve convexity, namely $\sum_{i=1}^{k} w_i = 1$

Question: How can non convexity affect the weighted kNN method?

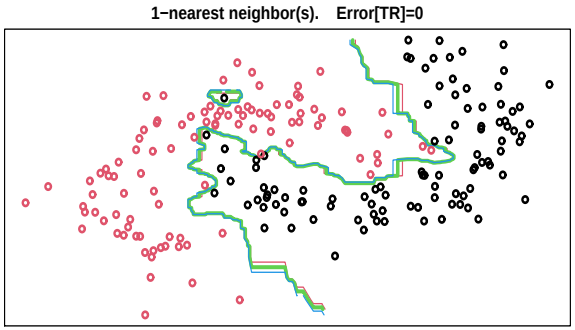
Effect of k on kNearest Neighbors Classification

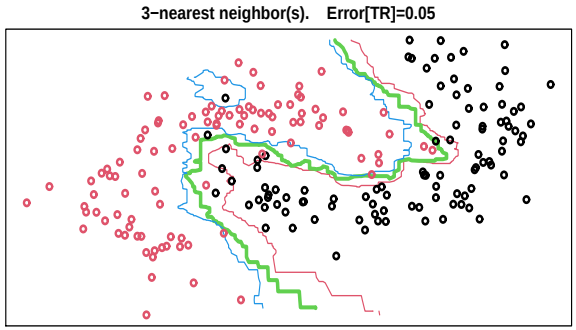
1. *If k is small, ie the estimated class of $\mathbf{x}$ is determined based on very few neighbors, the resulting kNN classifier will have very low bias, but very high variance. If fact, in the limit, if k=1, the decision boundary will perfectly separate the classes on the training set (think of Mr Memory), but will perform poorly on the test set*
2. *If k is large, ie the estimated class of $\mathbf{x}$ is determined based on very many neighbors from far and wide, the resulting kNN classifier will have very large bias, but very low variance. In fact, for truly large k, the decision boundary will be a constant hyperplane*
3. *We see the need for the determination of an optimal k, one that achieves the trade-off between bias and variance.*
    1. *Determine k by cross validation*
    2. *Determine k by direct minimization of the estimated prediction error via a suitably chosen test set*

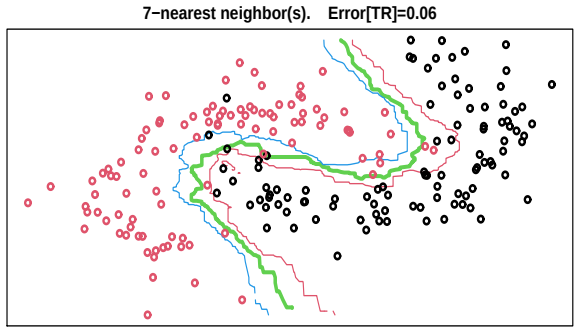
Various Effects in kNearest Neighbors Classification

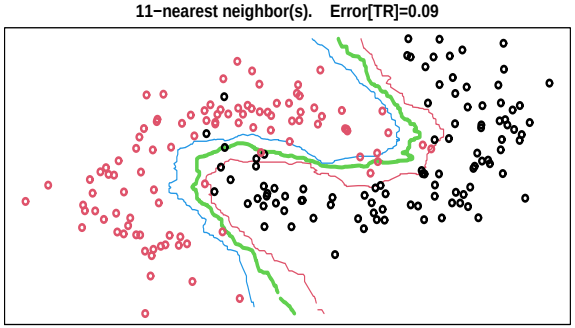
1. *Effect of k: It is now clear that in kNN, k controls the complexity of the underlying classifier , with small k yielding very complex classifiers and large k yielding rather simple ones*
2. *Effect of n: The sample size n plays a crucial role because kNN being a lazy method, all happens at prediction. A large n would therefore lead to intense prediction*
3. *Effect of p: The dimensionality p of the input space is only felt by the function that computes the distances. If the function is optimized, kNN should be unaffected by this dimensionality*
4. *Effect of distance: It is known that some distances are more robust to extreme observations*

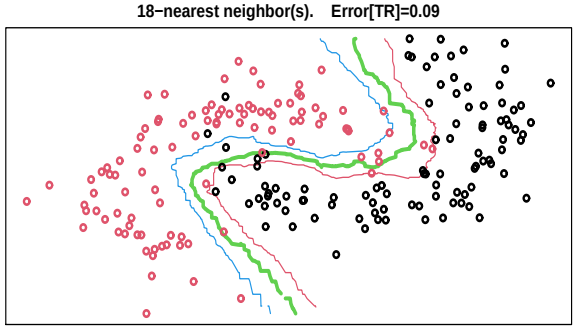
Effect of k on Decision Boundaries

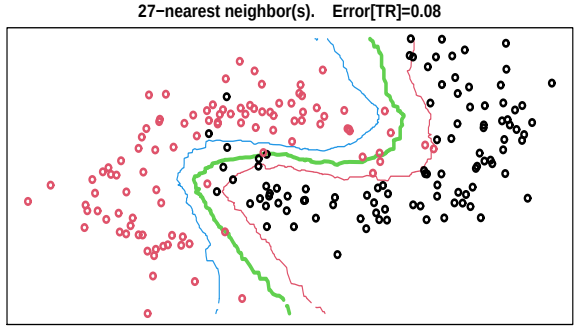


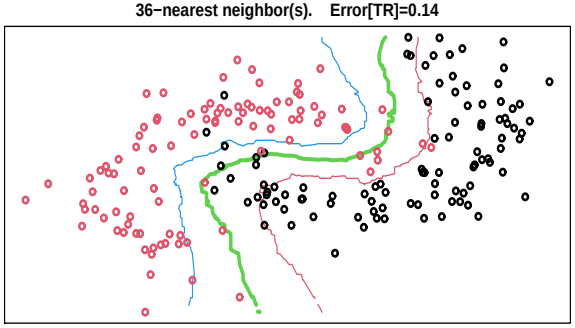


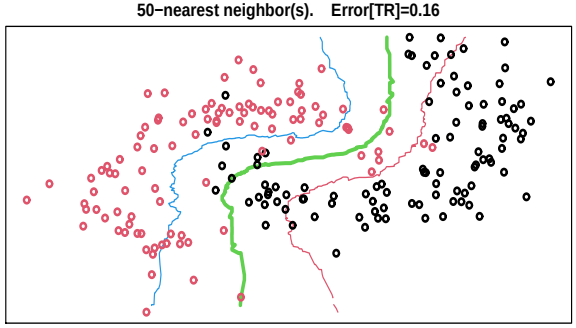


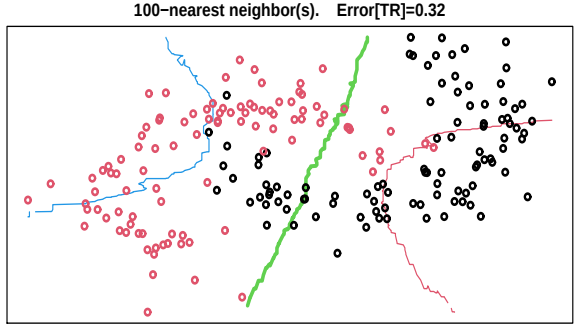






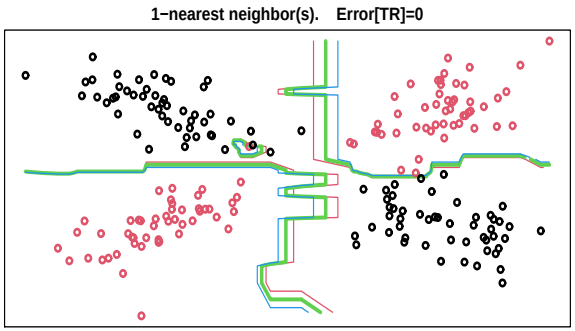


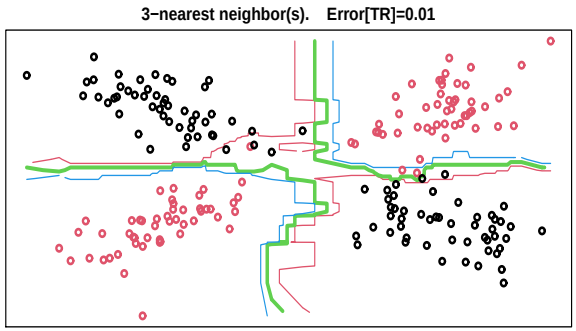


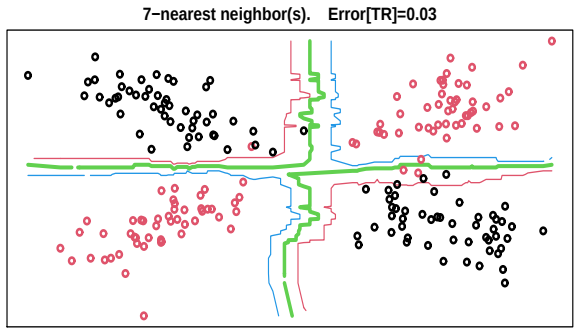


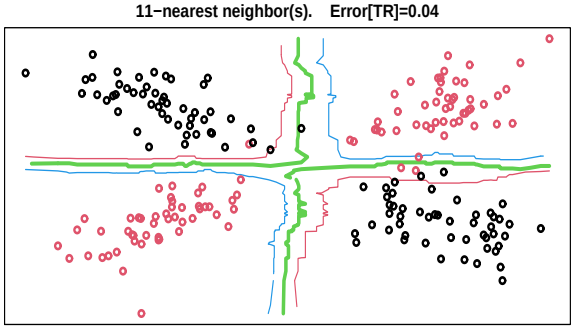
Various decision boundaries on the banana shape data set

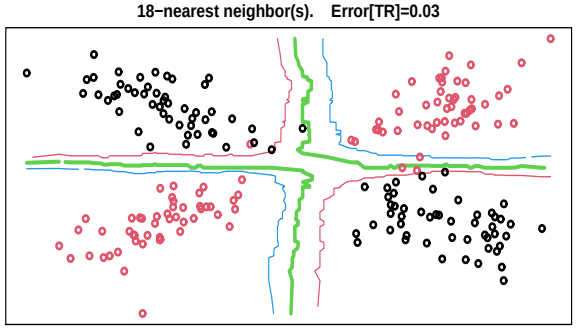
Effect of k on Decision Boundaries

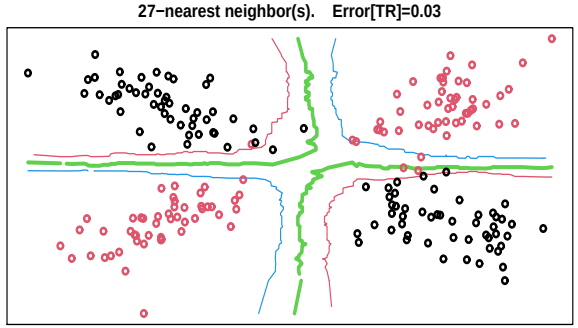


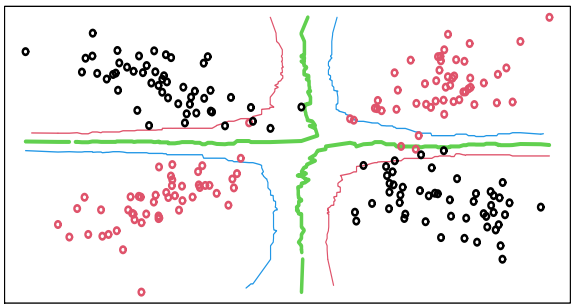


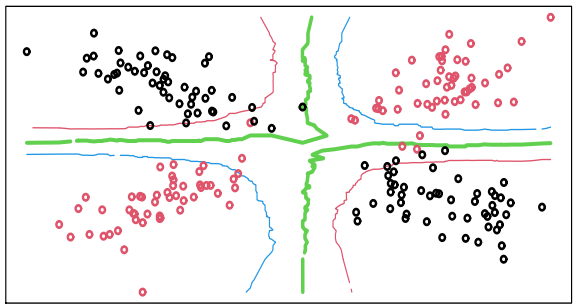


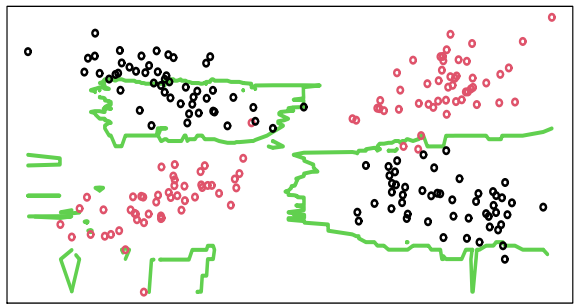












Various decision boundaries on the four corners data set

Distinct Strengths/Pros of Nearest Neighbors Approach

1. *The kNN method is intuitively appealing and very easy to understand, explain, program/code and interpret*
2. *The kNN method provides a decent estimate of* $\Pr[Y = g|\mathbf{x}]$*, the posterior probability of class membership*
3. *The kNN method easily handles missing values (by restricting distance calculations to subspace)*
4. *As the number of training samples grows larger, the asymptotic misclassification error rate is bounded by twice the Bayes risk.*

$$\lim_{n\to\infty} R(\hat{f}^{(n)}_{\texttt{kNN}}) \leq 2R^{\star}$$

where $R^{\star}$ *is the Bayes risk, that is, the small possible error, the error made by the generator of the data.*

5. *The kNN method is naturally suitable for sequential/incremental machine learning*
6. *The kNN method is also suitable where the hypothesis space is variable in size*

Distinct Strengths/Pros of Nearest Neighbors Approach

1. *The kNN method serves a basic and easy to understand foundational machine learning and data mining technique*
2. *The kNN method is an excellent baseline machine learning technique, and also allows many extensions*
3. *The kNN method can handle non-numeric data as long as the distance can be defined*
4. *The kNN method is usually performs reasonable well or sometimes very well when compared to more sophisticated techniques*
5. *kNN methods can handle mixed types of data as long as the distance are computed as hybrid or combinations*
6. *The kNN method is inherently multi-class, and this is very important because for some other methods, going beyond binary classification requires some sophisticated mathematics. It also handles very flexible decision boundaries*

Distinct Weaknesses/Cons of Nearest Neighbors Approach

1. *The computational complexity of kNN is very high in prediction. Specifically, it is $\mathcal{O}(nmp)$ where n is the training set size, m is the test set size and p is the number of predictor variables. This means that kNN requires large amount of memory, and therefore does NOT scale well. This failure in scalability is address using various heuristics and strategies*
2. *kNN methods suffer from the Curse of Dimensionality (COD). When p is large and n is small, especialy in the context of the so-called short fat data where $p \ggg n$,*
   - *The concept of nearness becomes meaningless to the point of being ill-defined when the dimension of the space is very high, because the "neighborhood" becomes very large*
   - *In a sense, one's nearest neighbor could be very far when the space is high dimensional and there are very few observations*

Distinct Weaknesses/Cons of Nearest Neighbors Approach

1. *The kNN method does not yield a model, and therefore no parameter to help explain why the method performs as it does*
2. *The kNN method is heavily affected by the local structure*
3. *The kNN method is very sensitive to both irrelevant and correlated features*
4. *Unless the distance is well chosen and properly calibrated, kNN methods will be sensitive to outliers and all sorts of noise in the data*
5. *Unless the distance is used in some way to weight the neighbors, more frequent classes will dominate in the determination of the estimated label. This means one has to be careful with kNN when one class has a far larger proportion of observations than the others*
6. *The measurement scale of each variable affect the kNN method more than most methods. This issue if usually tackled by simply standardizing, unitizing or cubizing/squeezing the data.*

Applications of the kNearest Neighbors Method

The k-Nearest Neighbors approach to Machine Learning and Data Mining has been successfully applied to wide variety of important fields. Amongst others:

1. *The famous and somewhat ubiquitous handwritten digit recognition (See example below from with data taken Hastie, Tibshirani and Friedman). This is usually the first task in some Data Analytics competitions.*
2. *More recently, text mining and specific topic of text categorization/classification has made successful use of kNearest Neighbors approach (See Assigned article)*
3. *Credit Scoring is another application that has been connected with k Nearest Neighbors Classification (See assigned Article)*
4. *Disease diagnostics also has been tackled using k Nearest Neighbors Classifiers*

Handwritten Digit Recognition

Incidence of Diabetes among Pima Indian Women

Motivating Example: *A study originally published by the National Institute of Diabetes and Digestive and Kidney Diseases sought to determine the relationship between the incidence of diabetes in Pima Indian Women and some specific medical and personal characteristics. A sample of women at least 21 years old and of Pima Indian heritage living near Phoenix were chosen and tested for diabetes.*

`npreg`	*Number of pregnancies*
`glu`	*Plasma glucose concentration*
`bp`	*Diastolic blood pressure (mm Hg)*
`skin`	*Triceps skin fold thickness (mm)*
`bmi`	*Body mass index kg/m2*
`ped`	*Diabetes pedigree function*
`age`	*Age (years)*

The response is type : Yes = diabetic; No = Non diabetic.

Incidence of Diabetes among Pima Indian Women

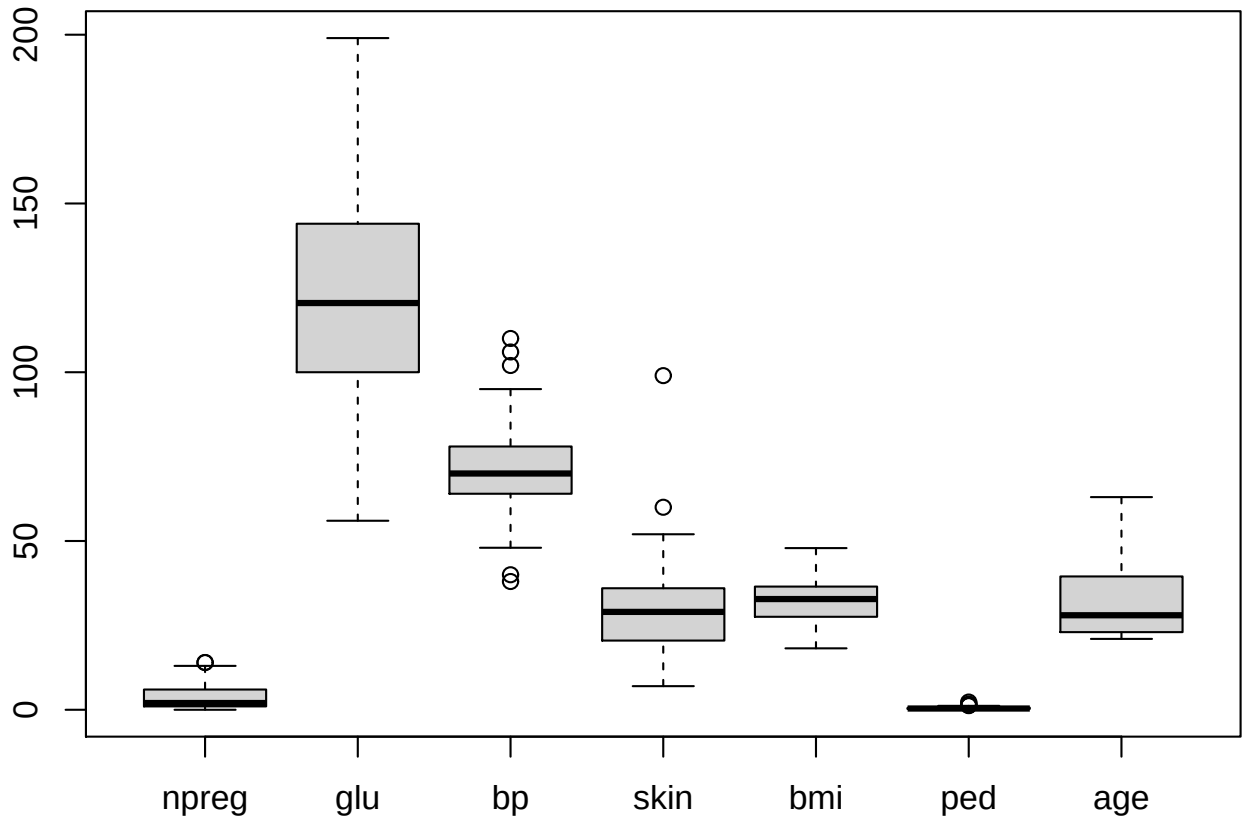
The dataset from the R package mlbench has $n = 200$ observations.

```
library(mlbench)
data(PimaIndiansDiabetes)          # load the data
xy <- PimaIndiansDiabetes          # Store data in xy frame
help(PimaIndiansDiabetes)          # learn stuff about this da
n   <- nrow(xy)                    # Sample size
```

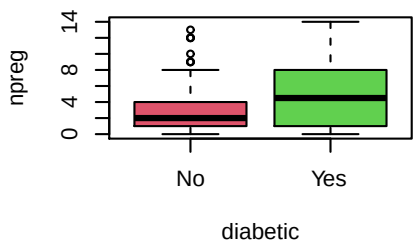
	npreg	glu	bp	skin	bmi	ped	age	type
1	5	86	68	28	30.20	0.36	24	No
2	7	195	70	33	25.10	0.16	55	Yes
3	5	77	82	41	35.80	0.16	35	No
4	0	165	76	43	47.90	0.26	26	No
5	0	107	60	25	26.40	0.13	23	No
6	5	97	76	27	35.60	0.38	52	Yes

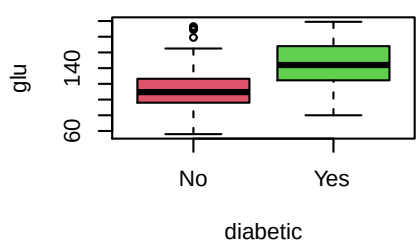
Pattern Recognition: How can kNearest Neighbors classification help build a predictively optimal classifier for this data?

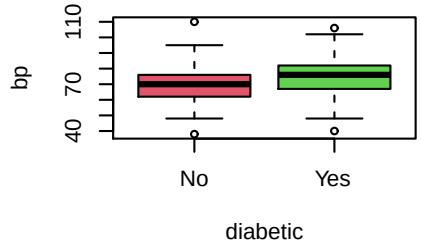
Analysis of the Pima Indian Diabetes Data

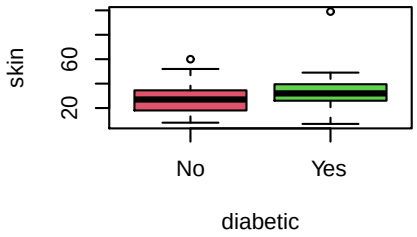


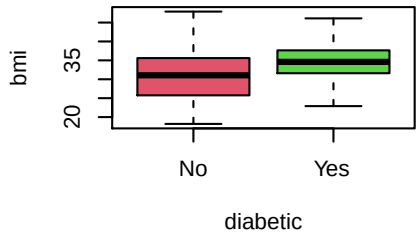
Analysis of the Pima Indian Diabetes Data

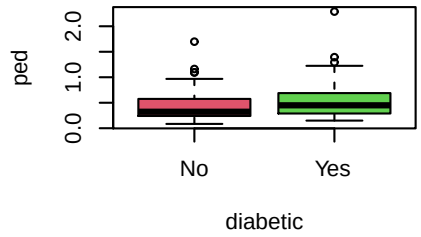


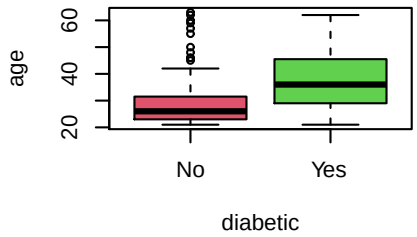




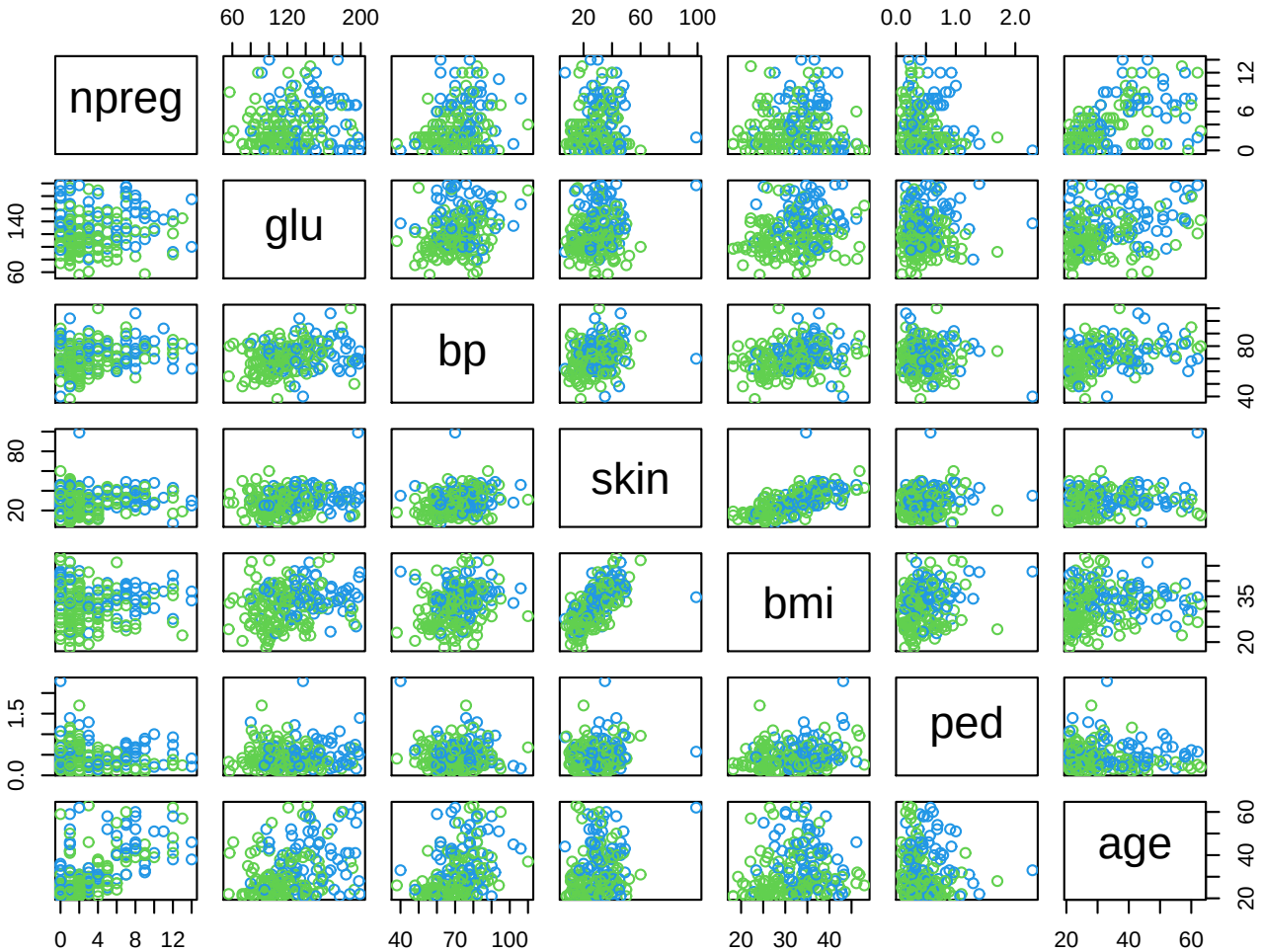




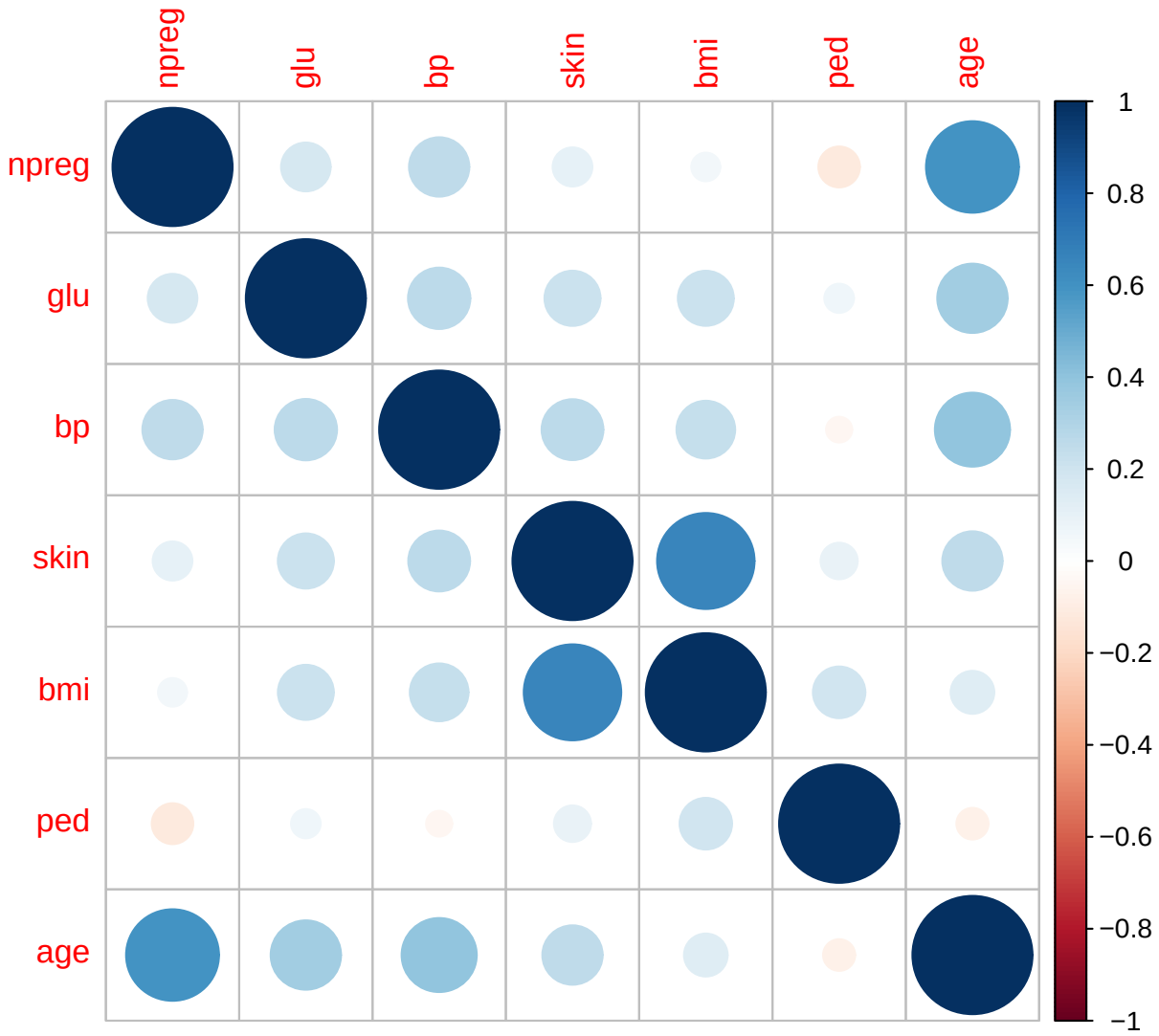




Analysis of the Pima Indian Diabetes Data



Analysis of the Pima Indian Diabetes Data

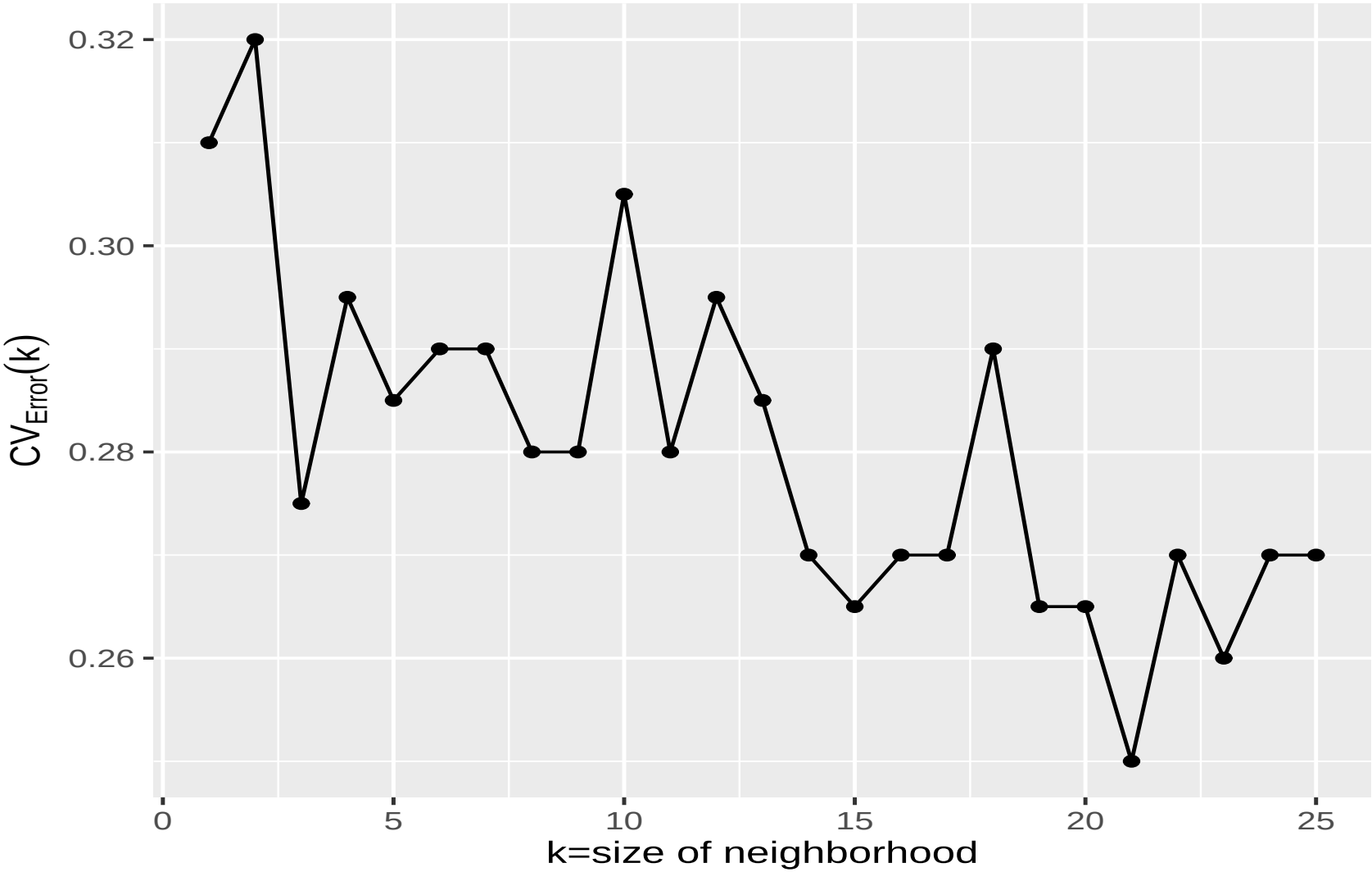


kNN Classification On the Pima Indian Diabetes Data

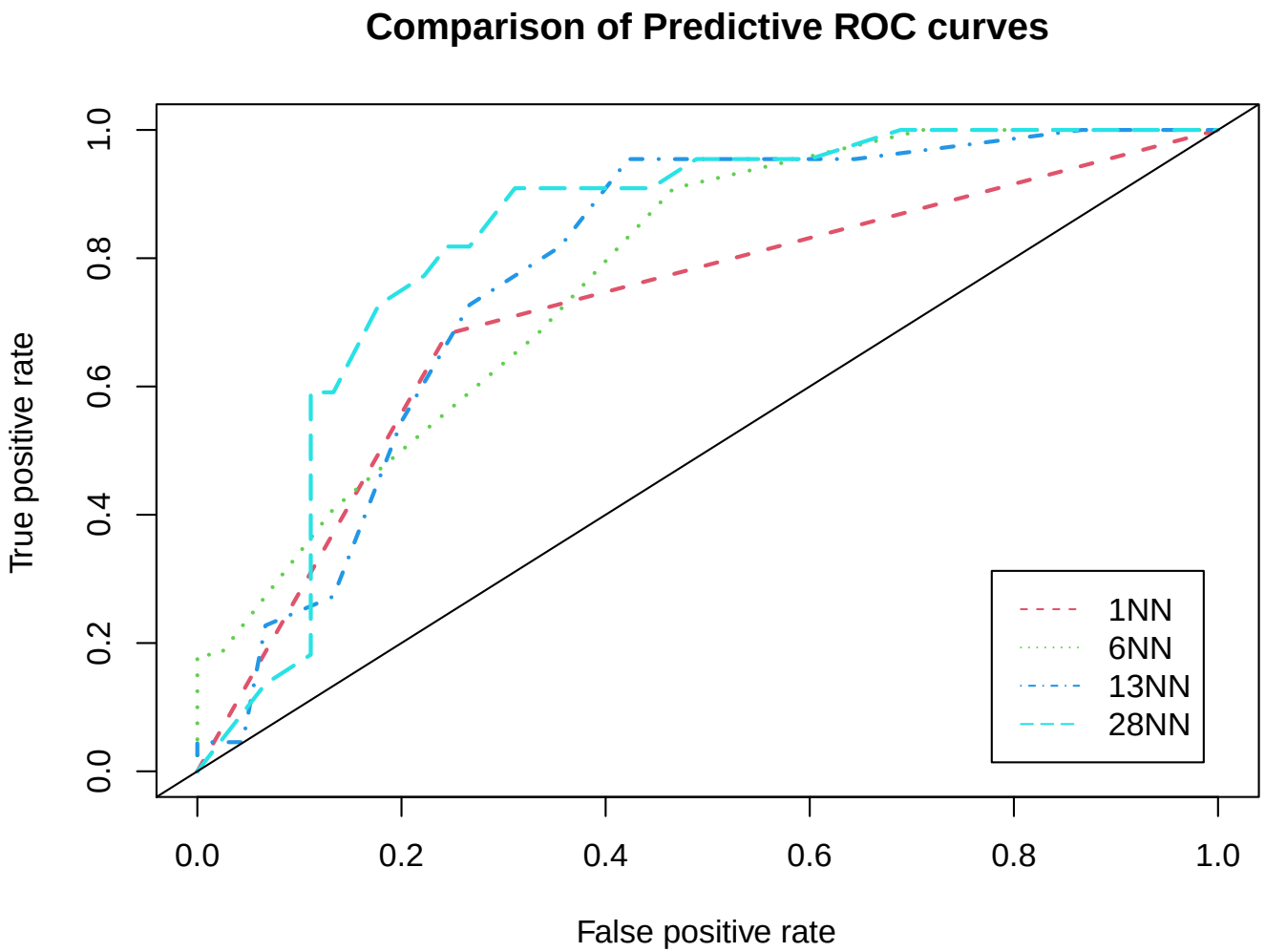
One of the greatest advantages of the kNN classifier is its simplicity, seen here on the famous Pima Indian Diabetes

```
library(mlbench)
library(class)
set.seed (19671210)   # Set seed for random number genera
epsilon <- 1/3        # Proportion of observations in the
nte     <- round(n*epsilon)  # Number of observations in
ntr     <- n - nte
id.tr   <- sample(sample(sample(n)))[1:ntr]   # For a sam
#id .tr <- sample(1:n, ntr, replace=F)        # Another w
id.te   <- setdiff(1:n, id.tr)
y.te         <- y[id.te]
y.te.hat    <- knn(x[id.tr,], x[id.te,], y[id.tr], k=9)  #
conf.mat.te <- table(y.te, y.te.hat
```

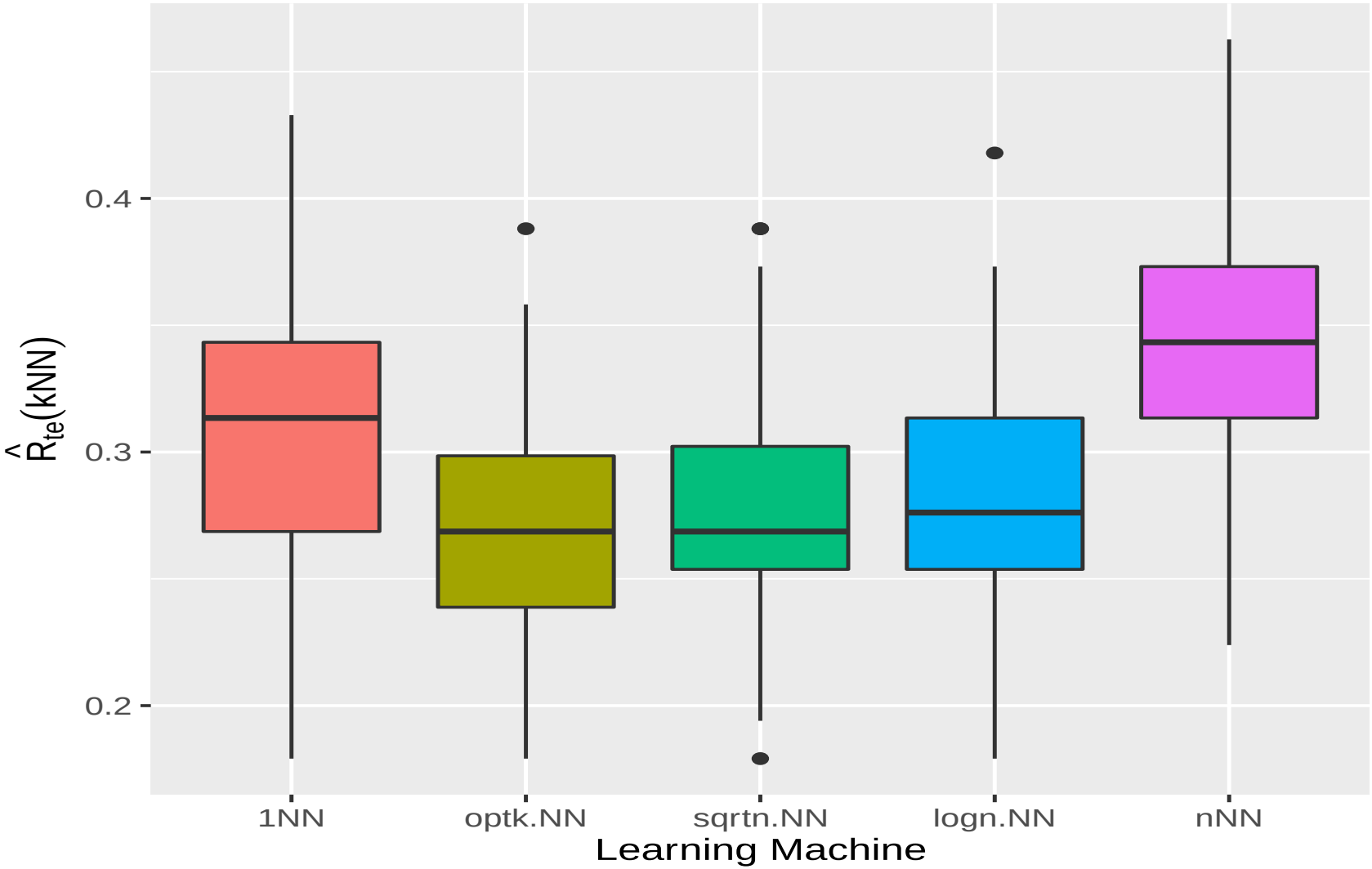
Analysis of the Pima Indian Diabetes Data



Analysis of the Pima Indian Diabetes Data

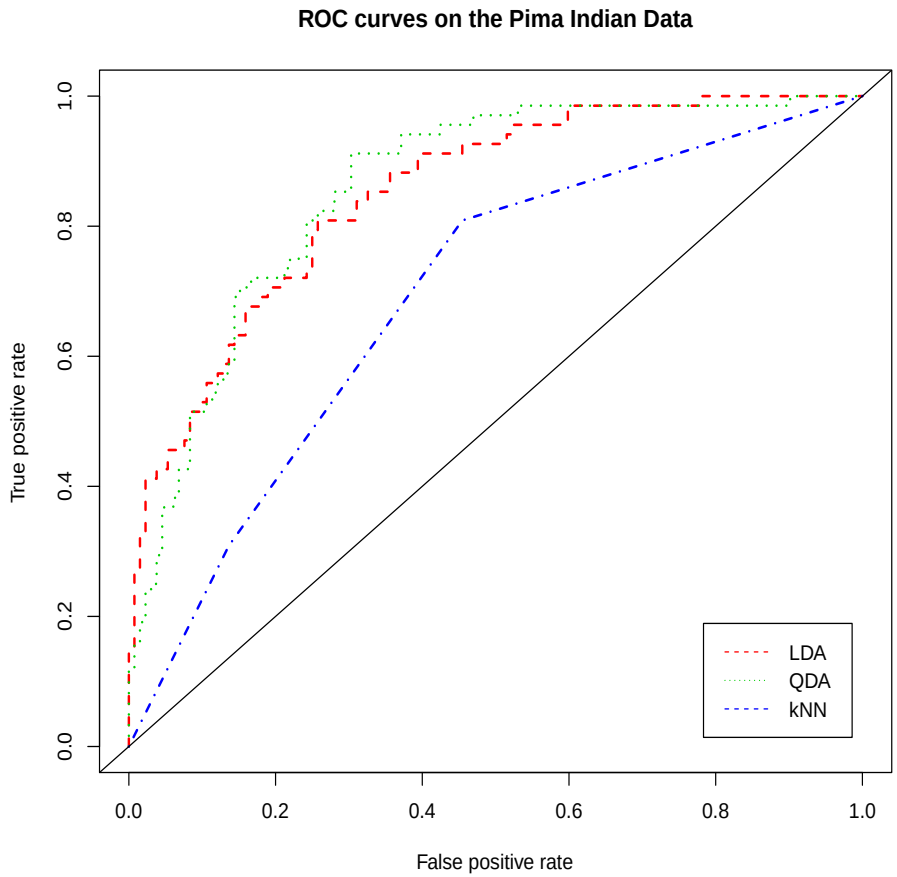


Analysis of the Pima Indian Diabetes Data



kNN vs others on Pima Indian Data

Below is a comparative ROC curve on the Pima Indian Data



Clearly, kNN looks inferior on this data. But ... Can it be improved? Different distance? Better k?

Exercises and Problems

1. *Consider the kNearest Neighbors learning machine with $k = 1$, also known as 1NN or NN learning machine. Given a sample of size n, namely $\mathscr{D}_n = \{(\mathbf{x}_i, \mathrm{y}_i),\ \mathbf{x}_i \in \mathscr{X}, \mathrm{y}_i \in \{1, 2, \cdots, G\}\}$,*
   - *Find the in sample prediction for $\mathbf{x}_{i_m}$, where $i_m = \lceil (n+1)/2 \rceil$, namely*

$$\widehat{f}_{\texttt{NN}}(\mathbf{x}_{i_m})$$

   - *Find the in-sample error rate under the zero-one loss*

$$\widehat{R}_n(\widehat{f}_{\texttt{NN}}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(\mathrm{y}_i \neq \widehat{f}_{\texttt{NN}}(\mathbf{x}_i))$$

   - *What is the bias of this learning machine?*

$$\texttt{Bias}(\widehat{f}_{\texttt{NN}})$$

Philosophical Basis of the Nearest Neighbors Paradigm

You are the average of your five people you spend the most time with.
Jim Rohn

You will end up being the average of your five closest friends.
Evan Carmichael

Take an inventory of your 5 closest friends. You will be an average of their influence.
Aaron Walker

Intuition of kNearest Neighbors Regression

- *kNearest Neighbors Principle:* *The reasonable expected value of any given object is most likely the typical (average/median/mode) value its nearest neighbors*

- *kNearest Neighbors Steps:* *Given a new point for which a prediction is sought,*
  - *Choose a distance for measuring how far a given point is from another*
  - *Set the size of the neighborhood k*
  - *Compute the distance from each existing point to the new point*
  - *Identify the values of the k points closest/nearest to the new point*
  - *Assign the average/median/mode of the response values of the neighbors as the best guess of the response value of the new point*

- *kNearest Neighbors Regression:* *The estimated response value associated with a vector $\mathbf{x}$ is the average of the response values in the neighborhood of $\mathbf{x}$.*

kNearest Neighbors (kNN) regression

$\mathscr{D}_n = \Big\{(\mathbf{x}_1, \mathrm{y}_1), \cdots, (\mathbf{x}_n, \mathrm{y}_n)\Big\}$, *with* $\mathbf{x}_i = (\mathrm{x}_{i1}, \cdots, \mathrm{x}_{ip})^\top \in \mathscr{X}$, $Y_i \in \mathbb{R}$.

1. *Choose the value of* k *and the distance to be used*
2. *Let* $\mathbf{x}^*$ *be a new point. Compute*

$$d_i = d(\mathbf{x}, \mathbf{x}_i) \qquad i = 1, \cdots, n$$

3. *Rank all the distances* d_i *in increasing order*

$$d_{(1)} \leq d_{(2)} \leq \cdots \leq d_{(k)} \leq d_{(k+1)} \leq \cdots \leq d_{(n)}$$

4. *Form* $\mathcal{V}_k(\mathbf{x})$*, the* k*-Neighborhood of* $\mathbf{x}$

$$\mathcal{V}_k(\mathbf{x}) = \{\mathbf{x}_j : \ d(\mathbf{x}, \mathbf{x}_j) \leq d_{(k)}\}$$

5. *Compute the predicted response* $\widehat{Y}$ *as*

$$\widehat{Y}_{\texttt{kNN}} = \widehat{f}_{\texttt{kNN}}(\mathbf{x}) = \frac{1}{k} \sum_{i=1}^{n} \mathrm{y}_j \mathbb{1}(\mathbf{x}_j \in \mathcal{V}_k(\mathbf{x}))$$

kNearest Neighbors (kNN) regression

Some properties of kNN estimators include

- *kNearest Neighbors (kNN) essentially performs regression by averaging the responses of the nearest neighbors of* $\mathbf{x}$.
- `kNN` *somewhat performs smoothing (filtering)*
- *The estimated response* $\widehat{Y}_{\texttt{kNN}}$ *for* $\mathbf{x}$ *is estimator of the average response which is the conditional expectation of* Y *given* $\mathbf{x}$

$$\widehat{Y}_{\texttt{kNN}} = \widehat{\mathbb{E}[Y|\mathbf{x}]}$$

- `kNN` *provides the most basic form of nonparametric regression*
- *Since the fundamental building block of* `kNN` *is the distance measure, one can easily perform regression beyond the traditional setting where the predictors are numeric. eg. Regression vectors of binary) indicator attributes*

$$\mathbf{x}_i = (\mathrm{x}_{i1}, \cdots, \mathrm{x}_{ip})^\top \in \{0,1\}^p$$

Algorithm 1 kNearest Neighbors (kNN) Learning Machine

Input: $\mathscr{D}_n = \{\mathbf{z}_i = (\mathbf{x}_i, \mathrm{y}_i),\ \mathbf{x}_i \in \mathscr{X}, \mathrm{y}_i \in \mathscr{Y},\ i = 1, \cdots, n\}$, neighborhood size k, sample size n, new test point $\mathbf{x}$, distance $d(\cdot, \cdot)$.

Output: Prediction $\widehat{f}_{(\texttt{kNN})}(\mathbf{x})$

for $i = 1, \cdots, n$ **do**

  Compute $d_i = d(\mathbf{x}, \mathbf{x}_i)$

Rank all the distances d_i in increasing order: $d_{(1)} \leq \cdots \leq d_{(k)} \leq \cdots \leq d_{(n)}$

Form $\mathcal{V}_k(\mathbf{x})$, the k-Neighborhood of $\mathbf{x}$: $\mathcal{V}_k(\mathbf{x}) = \left\{\mathbf{x}_i :\ d(\mathbf{x}, \mathbf{x}_i) \leq d_{(k)}\right\}$

Compute the predicted response $\widehat{f}_{(\texttt{kNN})}(\mathbf{x})$ as

$$\widehat{f}_{(\texttt{kNN})}(\mathbf{x}) = \begin{cases} \underset{g \in \mathscr{Y}}{\texttt{argmax}} \left\{ \dfrac{1}{k} \displaystyle\sum_{i=1}^{n} \mathbb{1}(y_i = g) \mathbb{1}(\mathbf{x}_i \in \mathcal{V}_k(\mathbf{x})) \right\} & \texttt{Classification} \\ \dfrac{1}{k} \displaystyle\sum_{i=1}^{n} y_i \mathbb{1}(\mathbf{x}_i \in \mathcal{V}_k(\mathbf{x})) & \texttt{Regression} \end{cases}$$

Exercises and Problems

1. *Consider the kNearest Neighbors learning machine with $k = n$, also known as nNN learning machine. Given a sample of size n, namely $\mathscr{D}_n = \{(\mathbf{x}_i, \mathrm{y}_i),\ \mathbf{x}_i \in \mathscr{X}, \mathrm{y}_i \in \mathbb{R}\}$,*
   - *Find the in sample prediction for $\mathbf{x}_{i_m}$, where $i_m = \lceil (n+1)/2 \rceil$, namely*

$$\widehat{f}_{\texttt{NN}}(\mathbf{x}_{i_m})$$

   - *Find the in-sample error rate under the squared error loss $\mathcal{L}(\mathrm{y}, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$, namely*

$$\widehat{R}_n(\widehat{f}_{\texttt{nNN}}) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(\mathrm{y}_i, \widehat{f}_{\texttt{nNN}}(\mathbf{x}_i))$$

   - *What is the variance of this learning machine?*

$$\texttt{Variance}(\widehat{f}_{\texttt{nNN}})$$

Multinomial of Statistical Machine Learning

- *Applications:* Sharpen your intuition and your commonsense by questioning things, reading about interesting open applied problems, and attempt to solve as many problems as possible
- *Methodology:* Read and learn about the fundamental of statistical estimation and inference, get acquainted with the most commonly used methods and techniques, and consistently ask yourself and others what the natural extensions of the techniques could be.
- *Computation:* Learn and master at least two programming languages. I strongly recommend getting acquainted with *R*

  http://www.r-project.org
- *Theory:* "Nothing is more practical than a good theory" (Vladimir N. Vapnik). When it comes to data mining and machine learning and predictive analytics, those who truly understand the inner workings of algorithms and methods always solve problems better.

References

[1] Clarke, B, Fokoué, E and Zhang, H (2009). *Principles and Theory for Data Mining and Machine Learning*. Springer Verlag, New York, (ISBN: 978-0-387-98134-5), (2009)