# STAT406- Methods of Statistical Learning Lecture 14

Matias Salibian-Barrera

UBC - Sep / Dec 2019

# Classification as prediction

- Most classifiers can be thought of as different ways to estimate or model

$$P\left(G = \mathbf{c}_j \middle| \mathbf{X} = \mathbf{x}\right)$$

- One way to model these probabilities is via Bayes' Thrm:

$$P\left(G = \mathbf{c}_j \middle| \mathbf{X} = \mathbf{x}\right) = \mathbf{f}\left(\mathbf{x} \middle| \mathbf{c}_j\right) P\left(G = \mathbf{c}_j\right) / \mathbf{f}\left(\mathbf{x}\right)$$

# LDA vs QDA

LDA classifies an observation into the class $c_j$ for which the estimated (under a normality assumption)

$$P\left(G = c_j \middle| \mathbf{X} = \mathbf{x}\right)$$

is highest.

If $\mathbf{X}|\, G = c_j \sim \mathcal{N}\left(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}\right)$ it means the class for which

$$\delta_j(\mathbf{x}) = \mathbf{x}'\,\mathbf{a}_j + b_j$$

is highest, where

$$\mathbf{a}_j = \boldsymbol{\Sigma}^{-1}\,\boldsymbol{\mu}_j \quad \text{and} \quad b_j = -\frac{1}{2}\,\boldsymbol{\mu}_j'\boldsymbol{\Sigma}^{-1}\,\boldsymbol{\mu}_j + \log(p_j)$$

# QDA

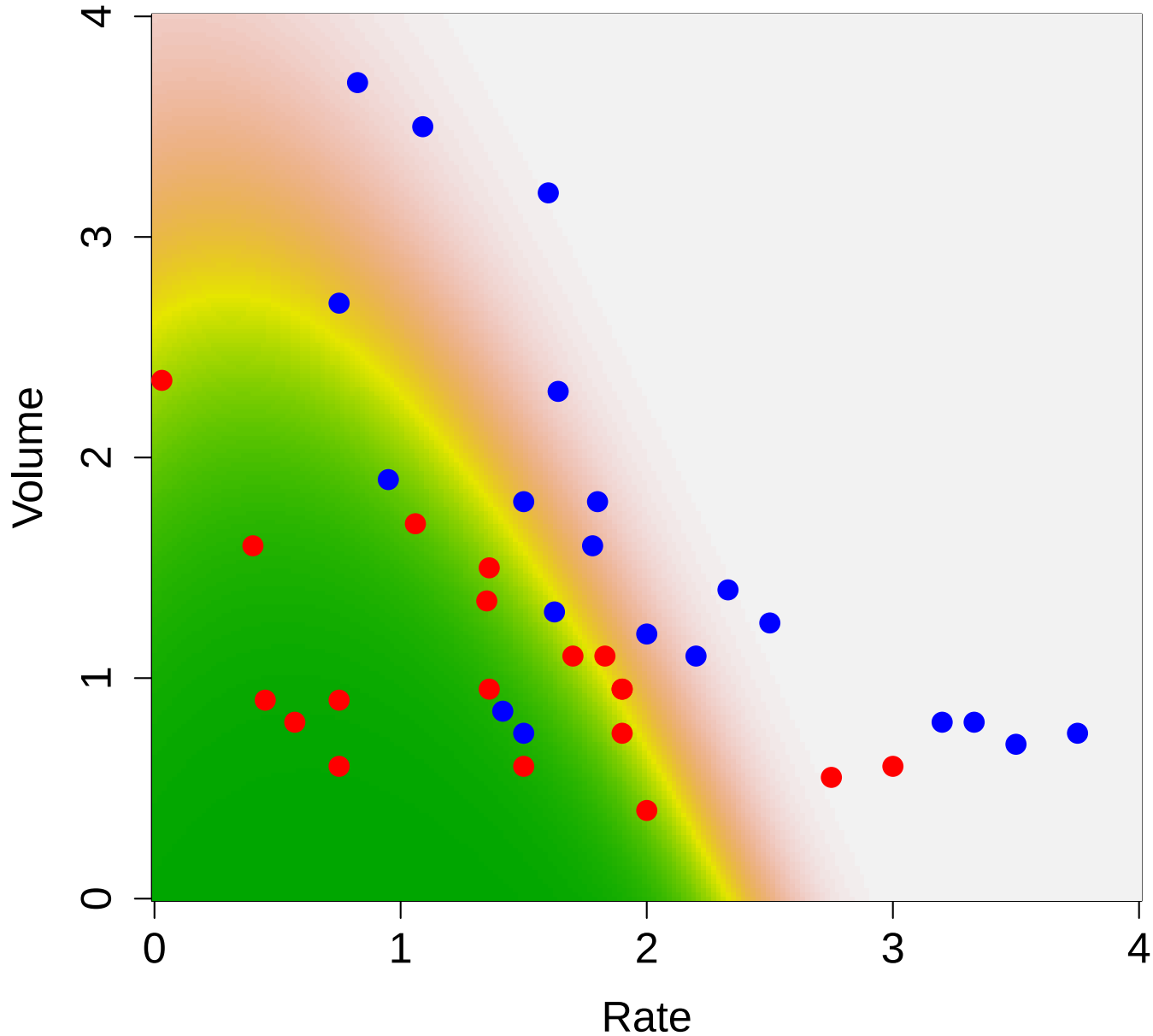
If $\mathbf{X}|\, G = c_j \sim \mathcal{N}\left(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)$ then, the estimated conditional (posterior) probabilities look different.

QDA classifies an observation into the class $c_j$ that maximizes

$$\delta_j(\mathbf{x}) = \mathbf{x}' \mathbf{A}_j \mathbf{x} + \mathbf{x}' \mathbf{d}_j + b_j$$

(see the book for details)

# QDA-based probabilities



# LDA - QDA - Logistic classifiers

- Multiclass: LDA, QDA and Logistic classifiers extend to the case of more than 2 classes

- Examples in R (Zip-code hand-written digits)

# LDA - QDA - Logistic classifiers

- LDA vs QDA: presents the usual "flexibility vs. variability" trade-off

- LDA & QDA vs Logistic classifiers: **Gaussian MLE estimates** (non-robust, sensitive to the Gaussian assumption) vs. **Binomial MLE estimates** (no distributional assumption of $\mathbf{X}|\, G = c_j$ required).

# Nearest-neighbours

- We need to estimate

$$P(G = \mathbf{g} | \mathbf{X} = \mathbf{x})$$

- An intuitive and “model-free” estimator is the nearest-neighbours estimator

# Nearest-neighbours

- Same spirit as the local-constant (kernel) regression estimator
- The K-NN estimator is

$$\hat{P}(G = \mathbf{g} | \mathbf{X} = \mathbf{x}) = \frac{1}{|N_{\mathbf{x}}^K|} \sum_{j \in N_{\mathbf{x}}^K} \mathbf{I}\{Y_j = \mathbf{g}\}$$

where

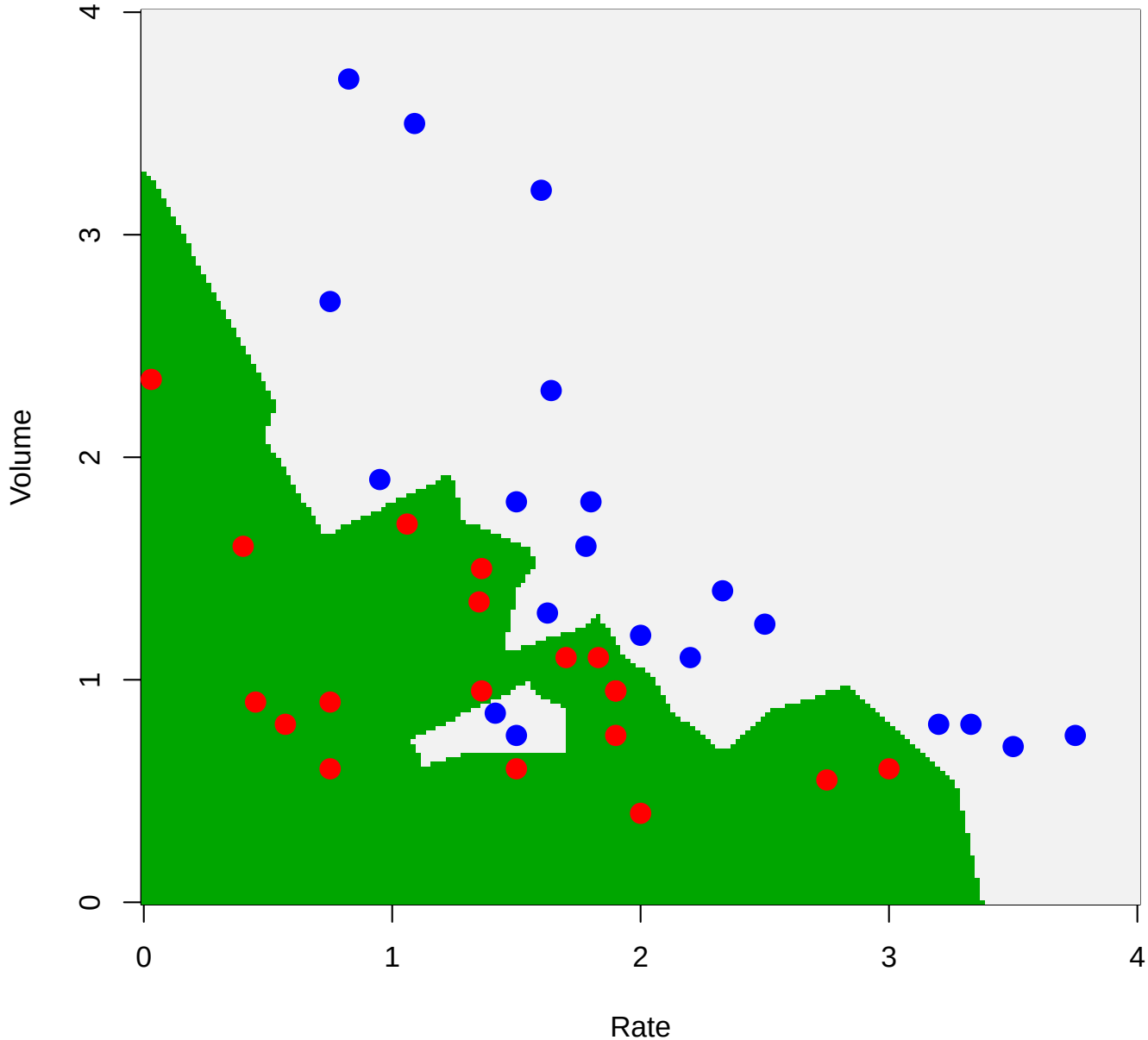
$$N_{\mathbf{x}}^K = \{i : d(\mathbf{X}_i, \mathbf{x}) \leq d_{(K)}\}$$

and $d_{(K)}$ is the distance from $\mathbf{x}$ to the $K$-th closest point in the sample ($\mathbf{X}_i$)

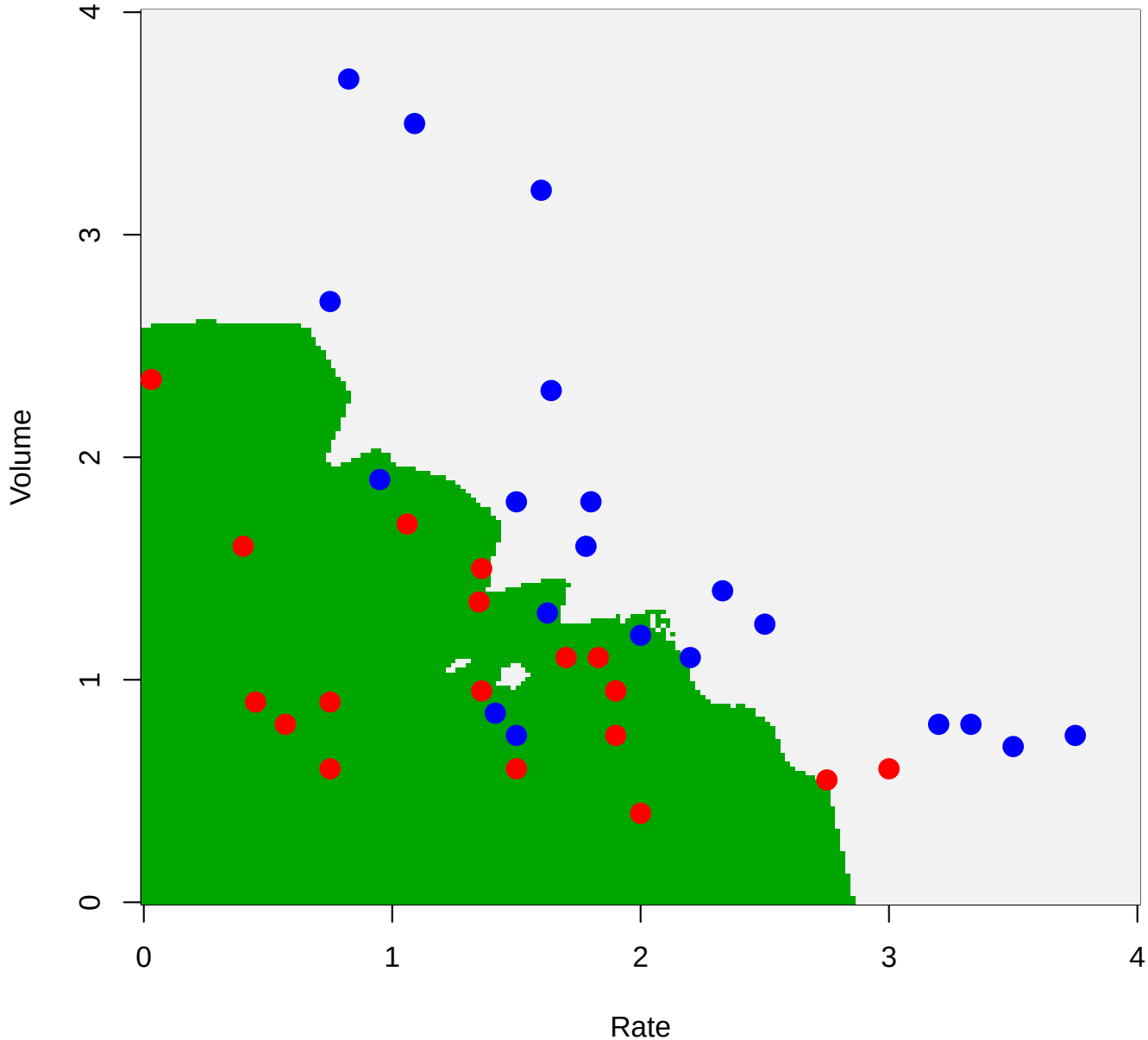
# Nearest-neighbours

- The K-NN estimator is the proportion of observations from class **g** among the closest *K* neighbours
- The K-NN classifier assigns a point to the class most represented among its K neighbours (“peer pressure”)

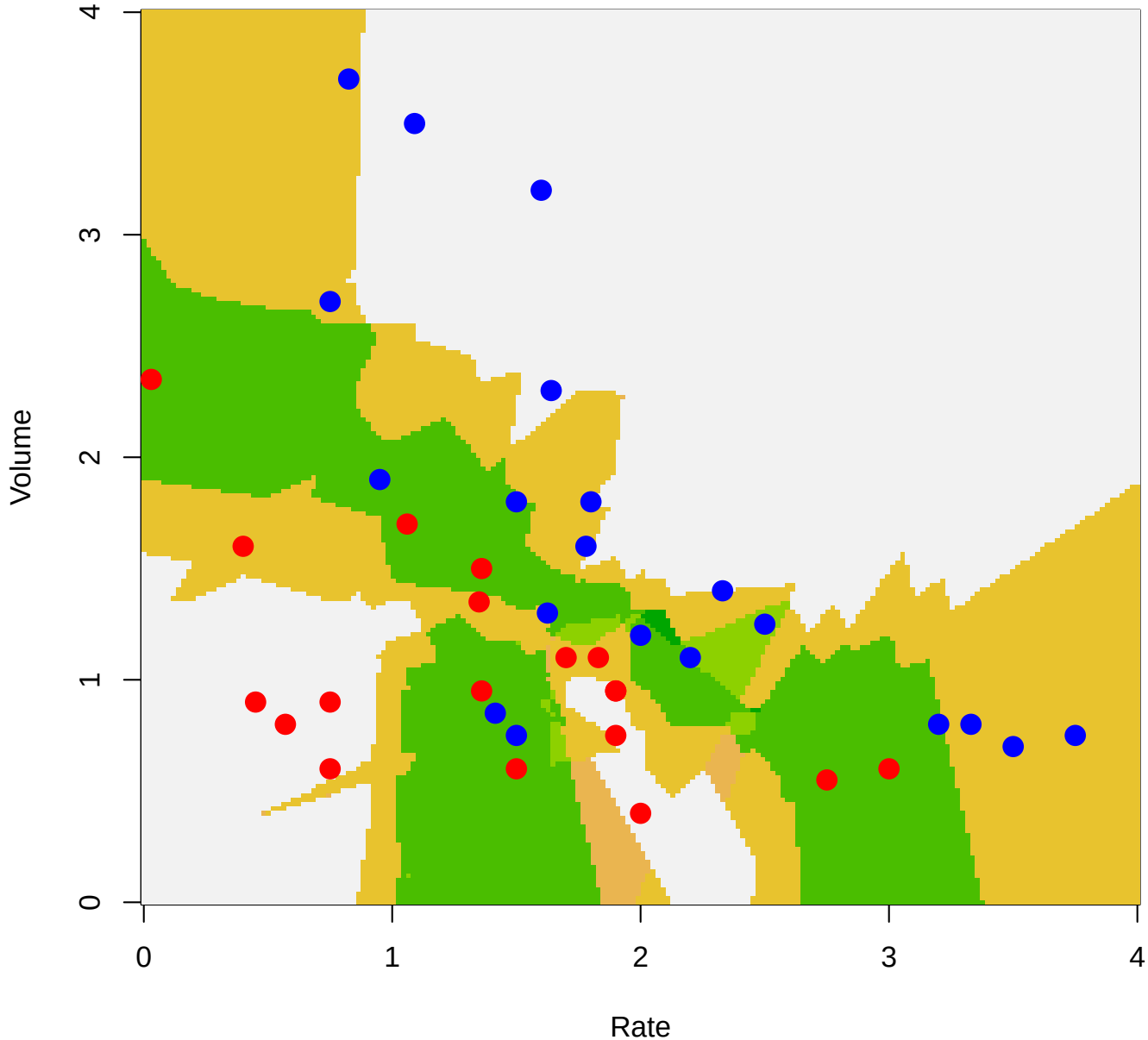
# Vaso – 1-NN



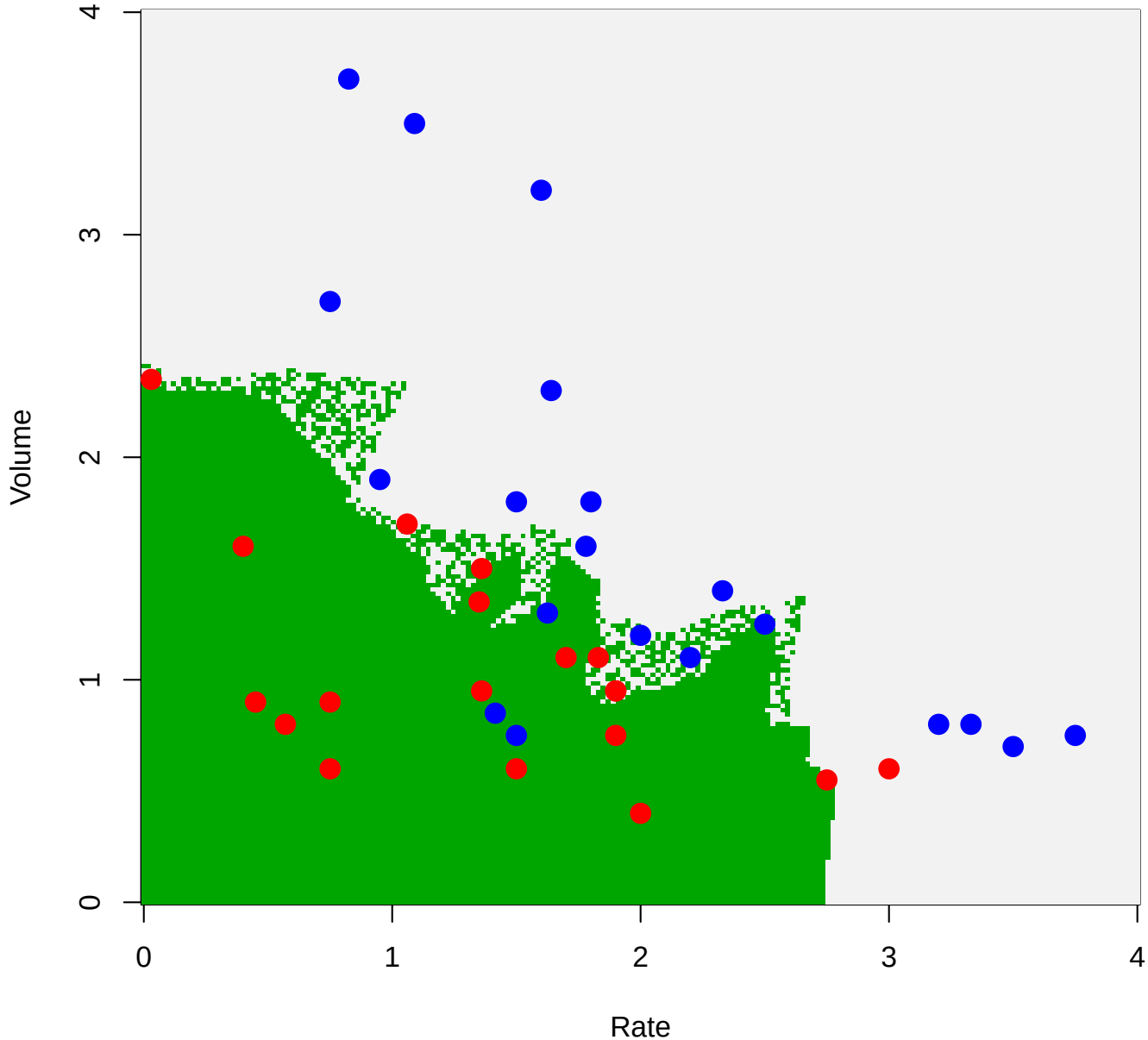
# Vaso – 5-NN



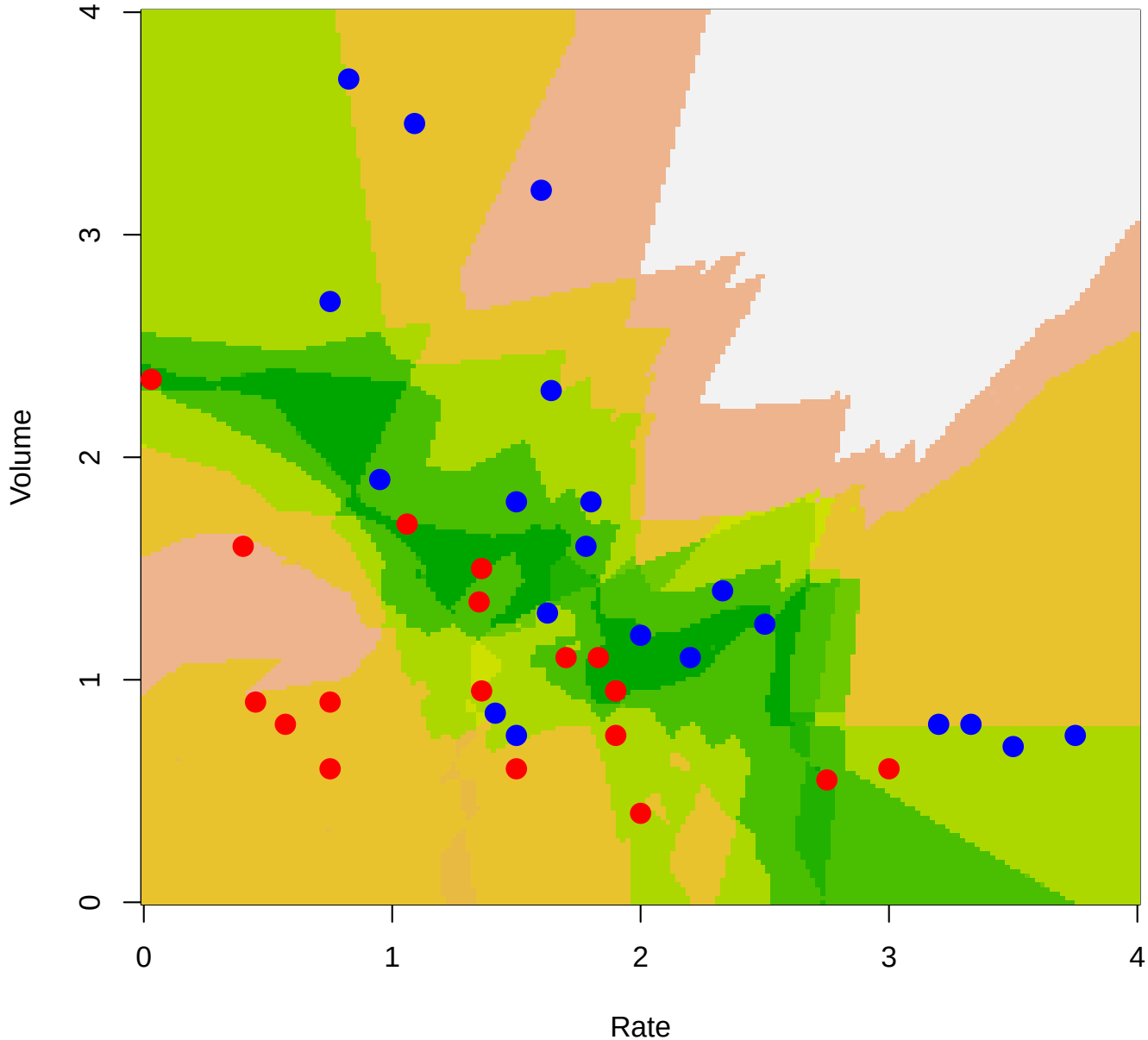
# Vaso – 5-NN - votes



# Vaso – 10-NN



# Vaso – 10-NN - votes



# Nearest-neighbours

- How can we select the number $K$ of neighbours?
- Zip-code example

# Nearest-neighbours - Challenge

- What's wrong with this picture?