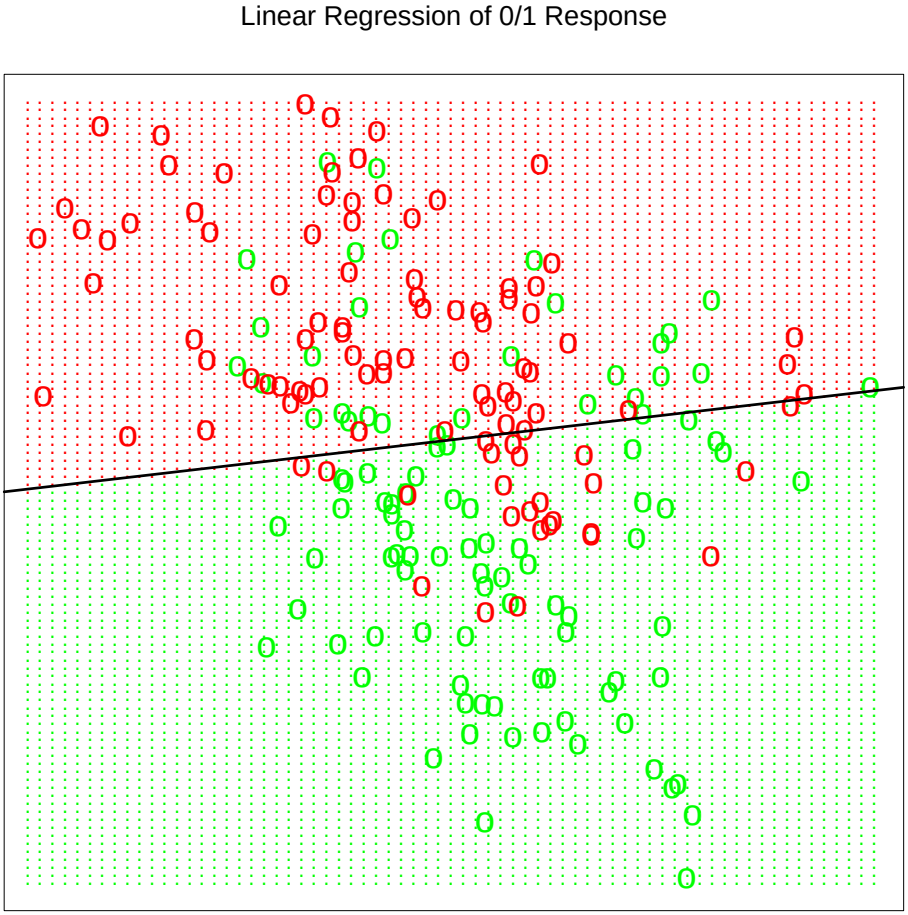

Figure 2.1: *A classification example in two dimensions. The classes are coded as a binary variable—* GREEN $= 0$, RED $= 1$*—and then fit by linear regression. The line is the decision boundary defined by* $x^T \hat{\beta} = 0.5$*. The red shaded region denotes that part of input space classified as* RED*, while the green region is classified as* GREEN*.*

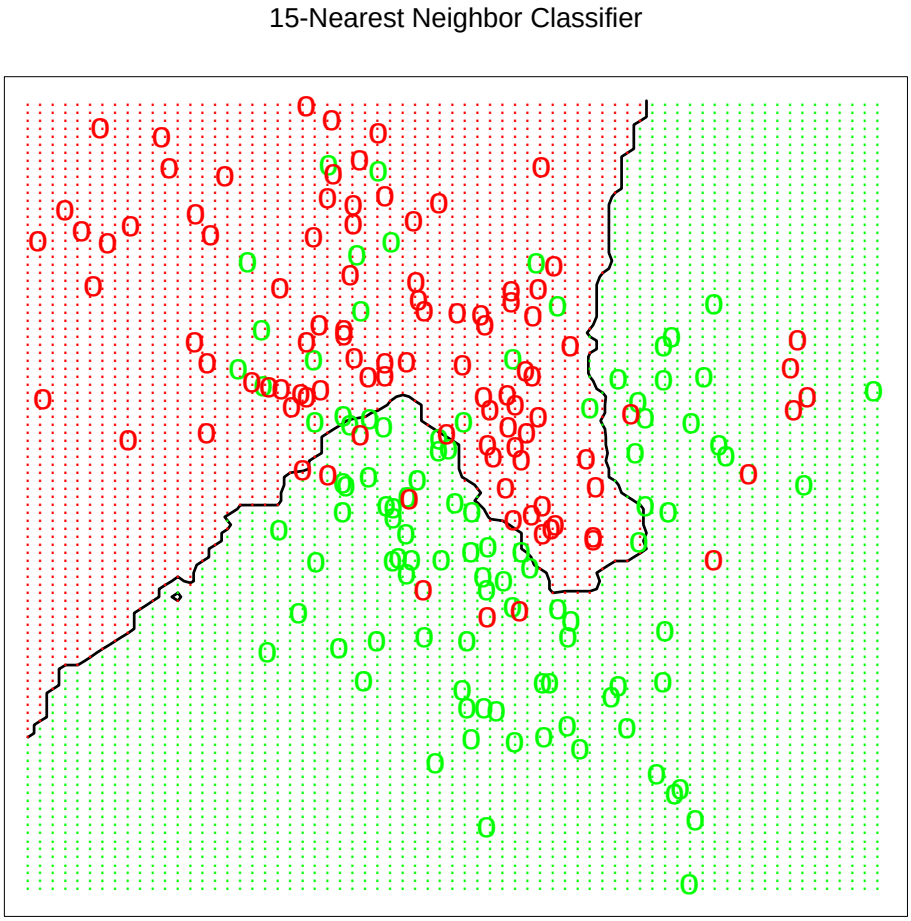

Figure 2.2: *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (GREEN $= 0$, RED $= 1$) *and then fit by* 15*-nearest-neighbor averaging as in (2.8). The predicted class is hence chosen by majority vote amongst the* 15*-nearest neighbors.*

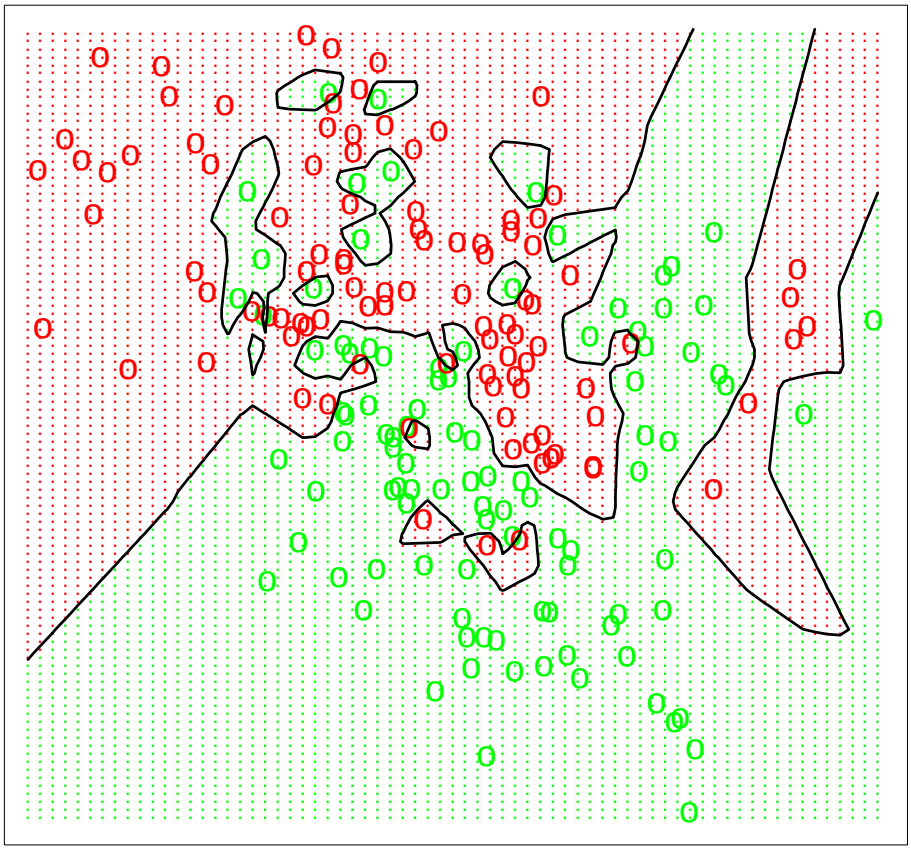

Figure 2.3: *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (GREEN $= 0$, RED $= 1$), *and then predicted by* 1*-nearest-neighbor classification.*

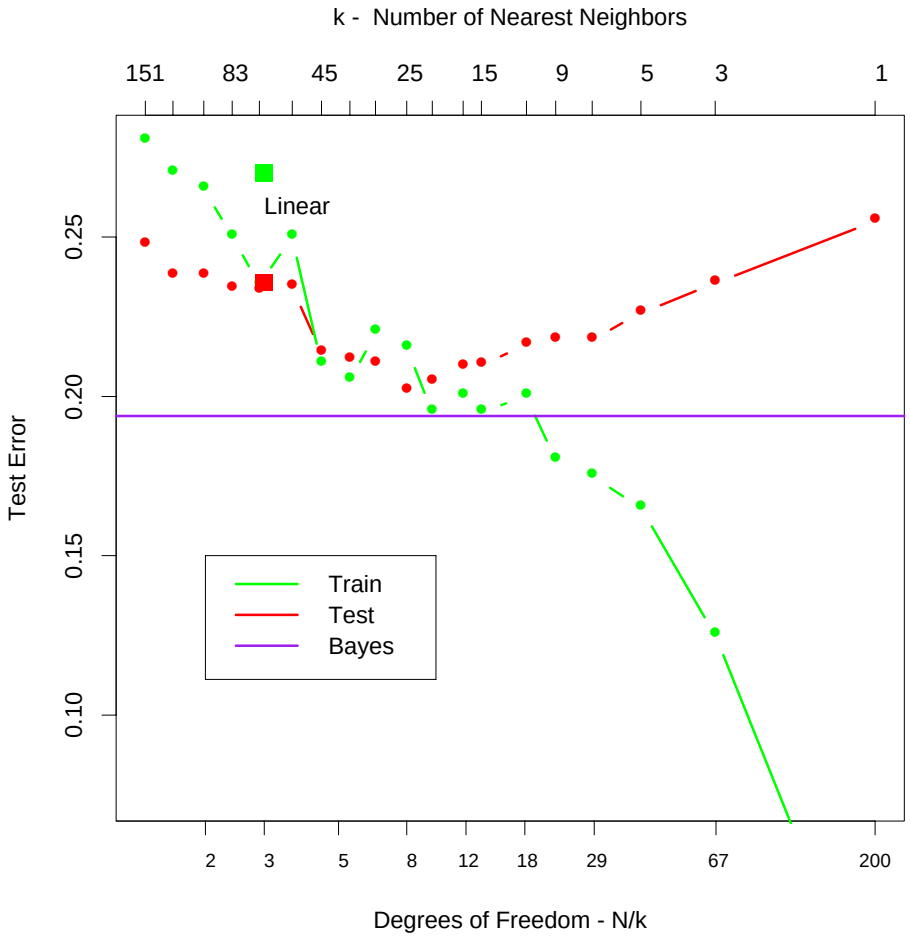

Figure 2.4: *Misclassification curves for the simulation example used in Figures 2.1, 2.2 and 2.3. A single training sample of size* 200 *was used, and a test sample of size* $10,000$. *The red curves are test and the green are training error for k-nearest-neighbor classification. The results for linear regression are the bigger green and red dots at three degrees of freedom. The purple line is the optimal Bayes Error Rate.*

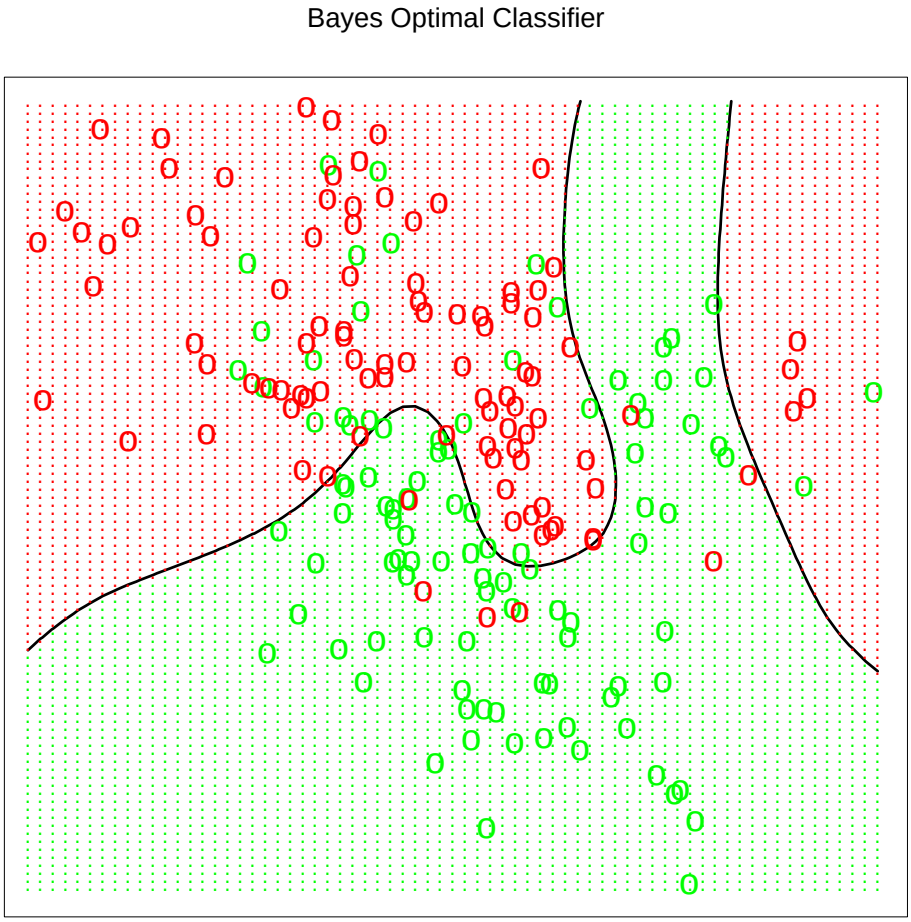

Figure 2.5: *The optimal Bayes decision boundary for the simulation example of Figures 2.1, 2.2 and 2.3. Since the generating density is known for each class, this boundary can be calculated exactly (Exercise 2.2).*

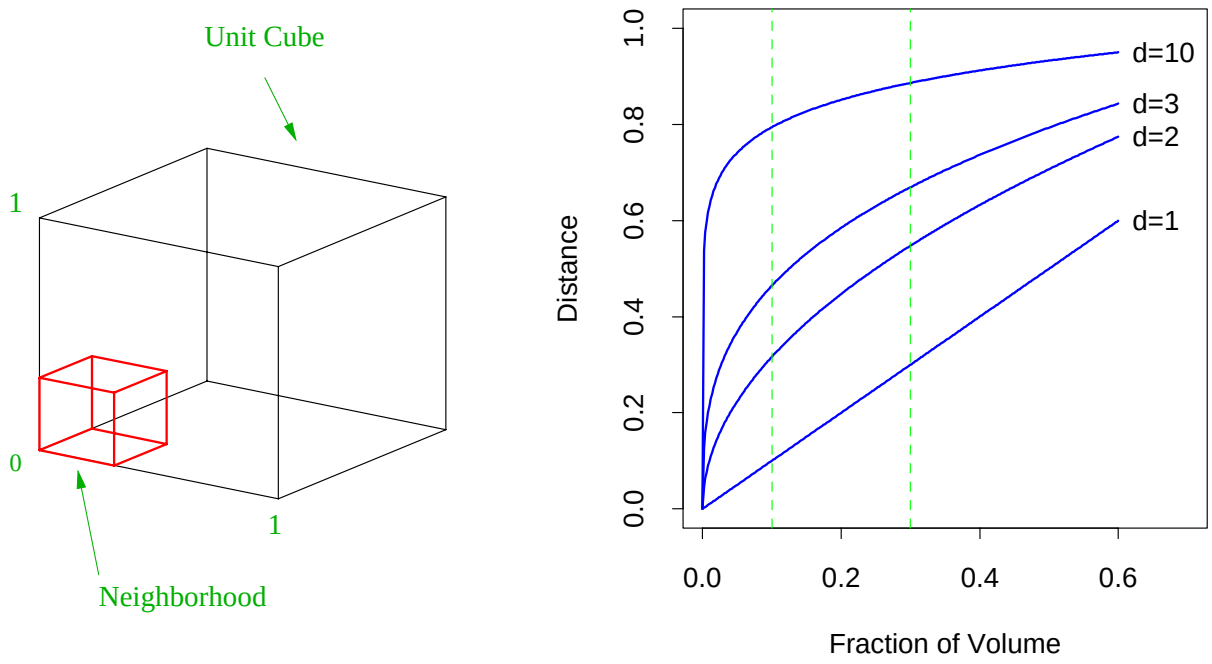

Figure 2.6: *The curse of dimensionality is well illustrated by a subcubical neighborhood for uniform data in a unit cube. The figure on the right shows the side-length of the subcube needed to capture a fraction r of the volume of the data, for different dimensions p. In ten dimensions we need to cover* 80% *of the range of each coordinate to capture* 10% *of the data.*

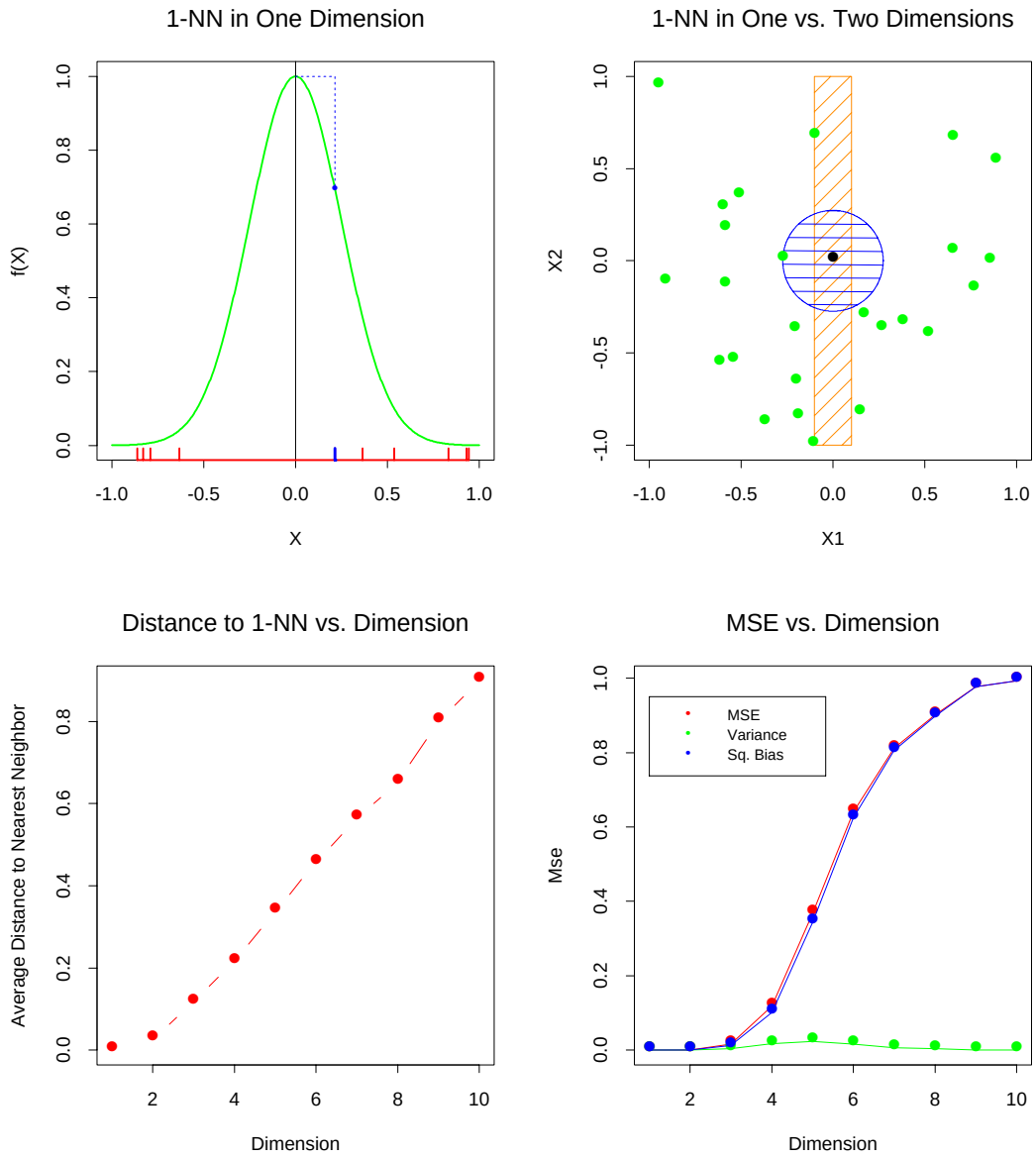

Figure 2.7: *A simulation example, demonstrating the curse of dimensionality and its effect on MSE, bias and variance. The input features are uniformly distributed in* $[-1,1]^p$ *for* $p = 1,\ldots,10$ *The top left panel shows the target function (no noise) in* $\mathbb{R}$*:* $f(X) = e^{-8||X||^2}$*, and demonstrates the error that* 1*-nearest neighbor makes in estimating* $f(0)$*. The training point is indicated by the blue tick mark. The top right panel illustrates why the radius of the* 1*-nearest neighborhood increases with dimension* p*. The lower left panel shows the average radius of the* 1*-nearest neighborhoods. The lower-right panel shows the MSE, squared bias and variance curves as a function of dimension* p*.*

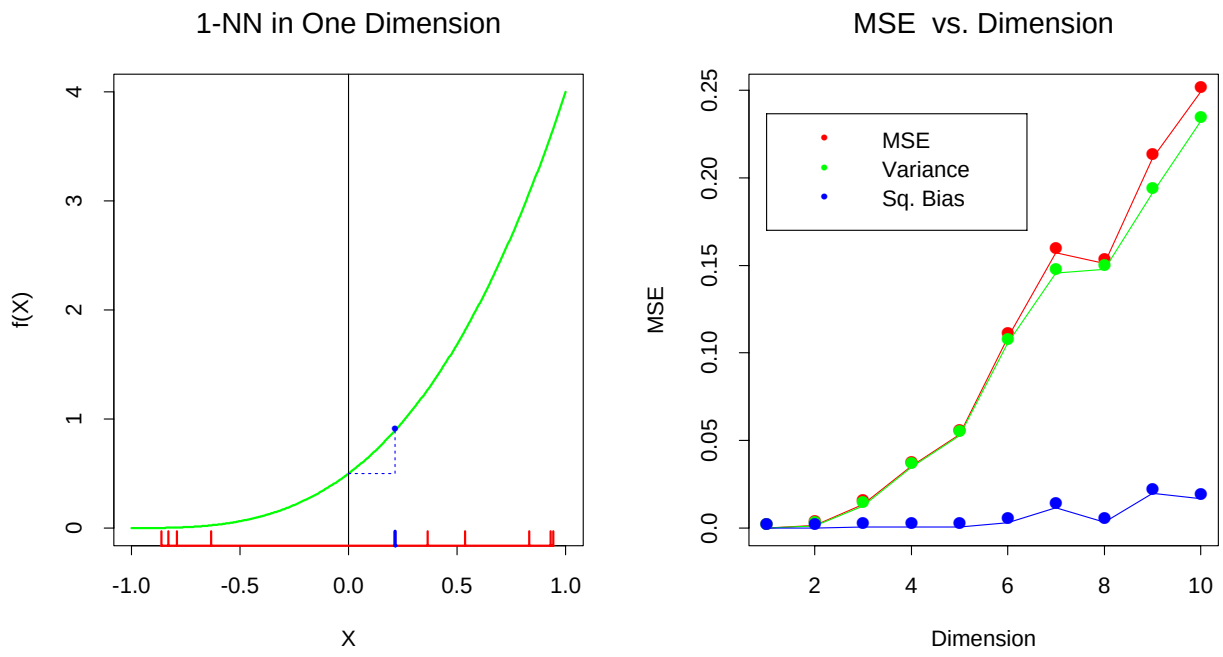

Figure 2.8: *A simulation example with the same setup as in Figure 2.7. Here the function is constant in all but one dimension:* $F(X) = \frac{1}{2}(X_1 + 1)^3$. *The variance dominates.*

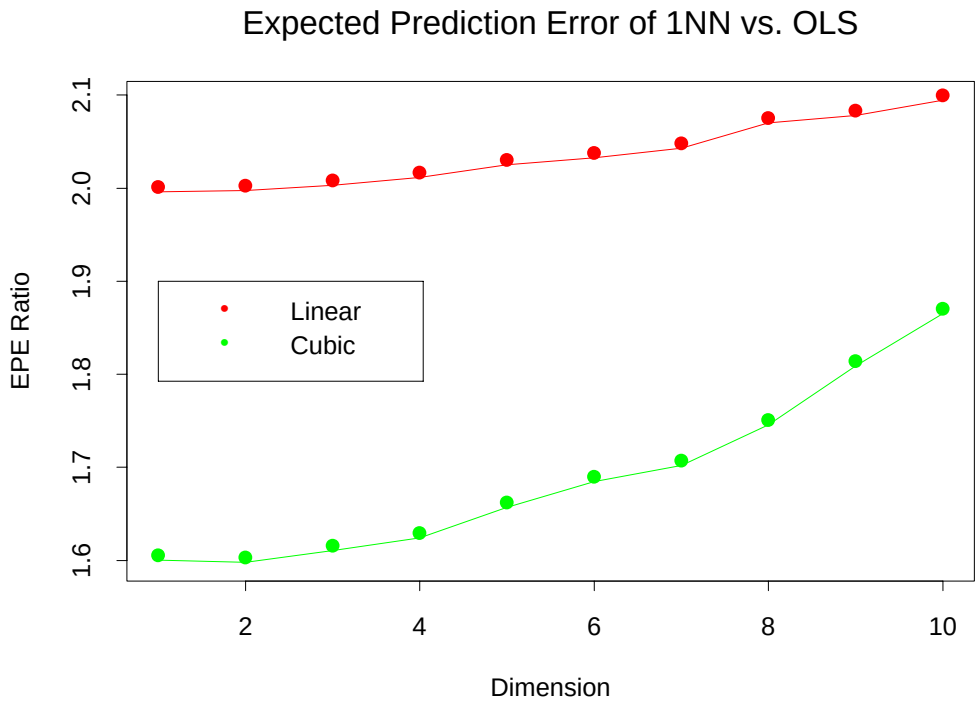

Figure 2.9: *The curves show the expected prediction error (at* $x_0 = 0$*) for* 1*-nearest neighbor relative to least squares for the model* $Y = f(X) + \varepsilon$*. For the red curve,* $f(x) = x_1$*, while for the green curve* $f(x) = \frac{1}{2}(x_1 + 1)^3$*.*

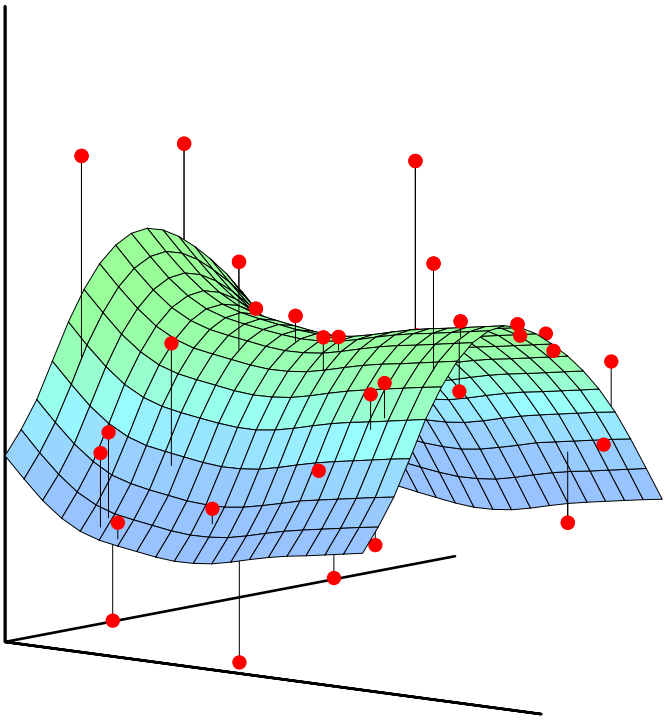

Figure 2.10: *Least squares fitting of a function of two inputs. The parameters of* $f_\theta(x)$ *are chosen so as to minimize the sum-of-squared vertical errors.*

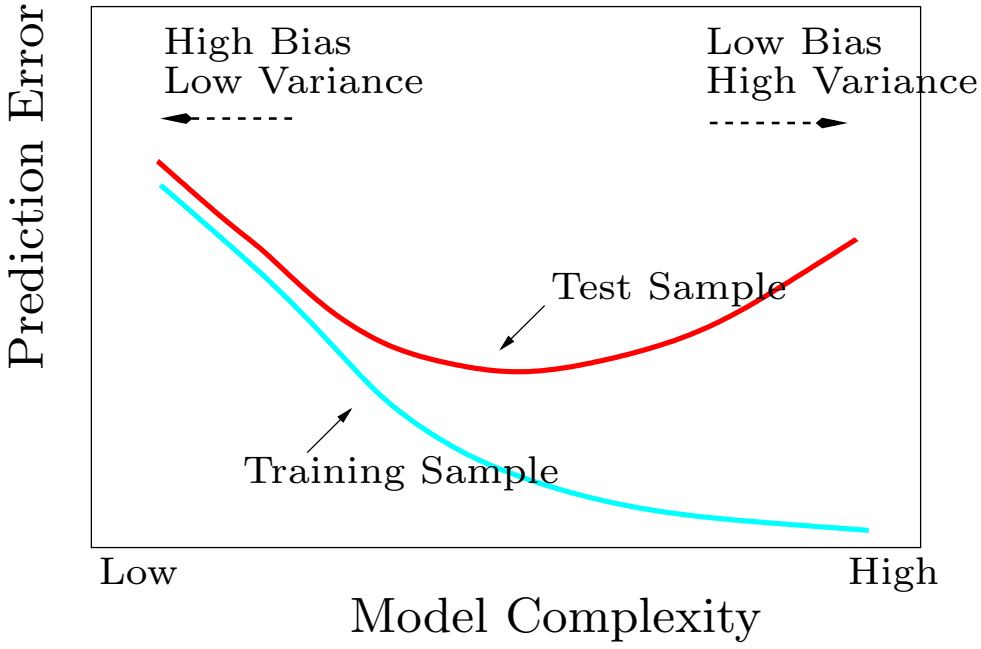

Figure 2.11: *Test and training error as a function of model complexity.*