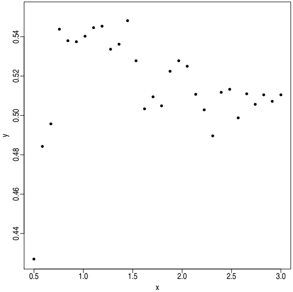
Bias/variance trade-off

A simple prototype problem

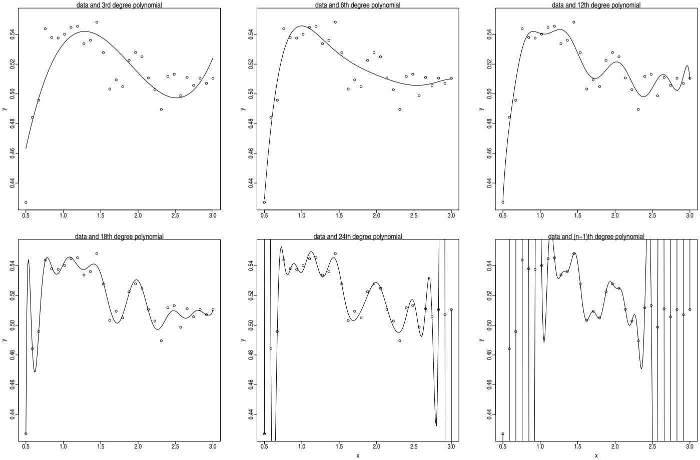


- Yesterday we observed n couples (x_i, y_i), for $i = 1, \ldots, n$, of data $(n = 30)$.
- These data are artificially generated by the law $y = f(x) + \text{error}$ where $f(x)$ is a unspecified smooth and regular function.
- We wish to obtain a rule (model), like $\hat{y} = \hat{f}(x)$, that enables us to predict y once we know x; a rule that allows us to predict y as new observations of x become available, say *tomorrow*.

A simple prototype problem

- a simple possibility is to interpolate data with a polynomial
- but, of which degree? $0, 1, 2, \ldots, 29$?
- Let's try to use polynomials of degree p (with $p = 0, 1, \ldots, n-1 = 29$).
  We need to estimate p parameters $(+\sigma^2)$.

A simple prototype problem

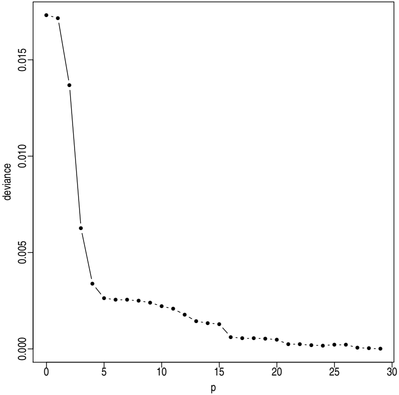


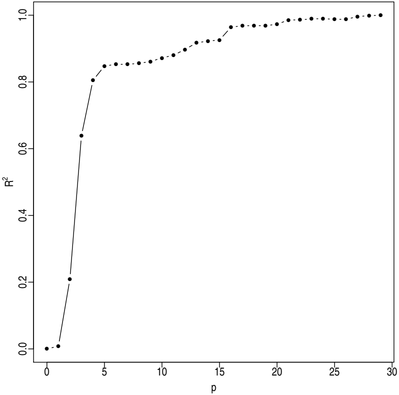
Source: AS2012

By growing of p the fitting of the polynomials is getting better.

A simple prototype problem

We measure the goodness of fit by obtaining, for each p the residual deviance and the coefficient of determination R^2.



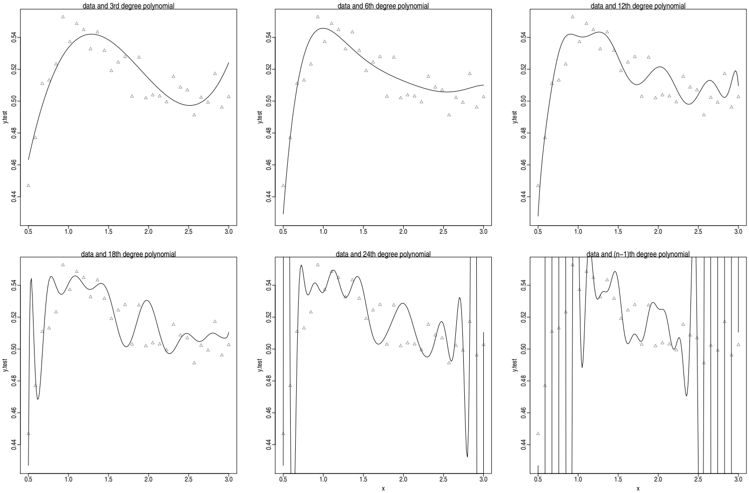


Source: AS2012

A simple prototype problem

- *Tomorrow* we will receive a new set of n data $\{y_i, i = 1, \ldots, n\}$, generated by the *same* phenomenon of the yesterday data, that is, the same function $f(x)$
- We want to predict these new observations, by assuming (for simplicity) that the new y_i are associated to the same x_i of the yesterday data.
- We compare our predictions (one for each polynomial) with the new data observed tomorrow.

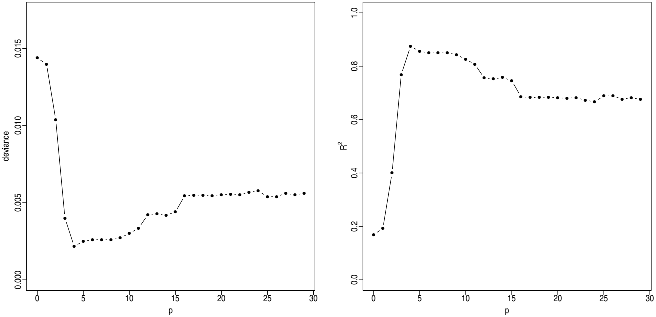
A simple prototype problem



Source: AS2012

A simple prototype problem

- Goodness of fit for each p: residual deviance and coefficient of determination R^2 on the new (*tomorrow*) data.



Source: AS2012

- Residual deviance first decreases, then increases, while R^2 reaches a maximum value and then decreases.

A simple prototype problem

If we knew $f(x)\dots$

- We want to estimate $f(x)$ using a generic estimator $\hat{y} = \hat{f}(x)$ (in our example, can be one of the 30 fitted polynomials)
- We start by considering a specific value x' of x, among the n observed.
- If we knew the mechanism used to generate the data precisely, we knew also $f(x')$, and we could calculate some quantities of interest to evaluate the estimator $\hat{y}$.
- For example, an important goodness-of-fit indicator is the mean squared error (with respect to the random variable y)

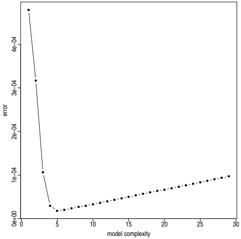
$$\mathbb{E}_y\{[\hat{y} - f(x')]^2\}$$

A simple prototype problem

- Since we are not interested only on the single point x', we consider the sum of the mean squared errors for all the n values of x,

$$\sum_{i=1}^{n} \mathbb{E}_y \left\{ \left[\hat{y} - f(x_i) \right]^2 \right\}$$

- If we do it for all the possible choices of p, which is an indicator of the model complexity, we may obtain the plot



Even if the true $f(x)$ is not a polynomial, there exists a degree p which is better than the others

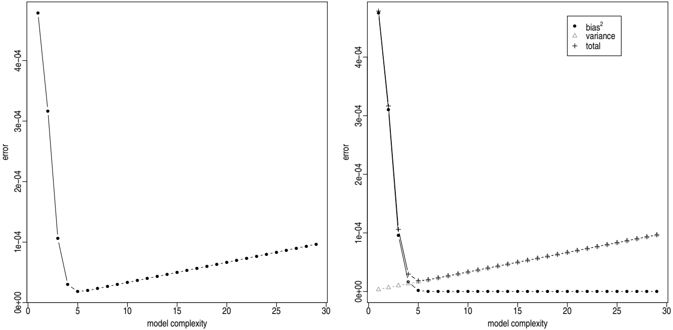
A trade-off

The *mean squared error* may be divided in two components

$$
\begin{aligned}
\mathbb{E}\big\{[\hat{y} - f(x')]^2\big\} &= \mathbb{E}\big\{[\hat{y} \pm \mathbb{E}\{\hat{y}\} - f(x')]^2\big\} \\
&= \big[\mathbb{E}\{\hat{y}\} - f(x')\big]^2 + \mathrm{var}\{\hat{y}\} \\
&= \text{bias}^2 + \text{variance}
\end{aligned}
$$

A trade-off

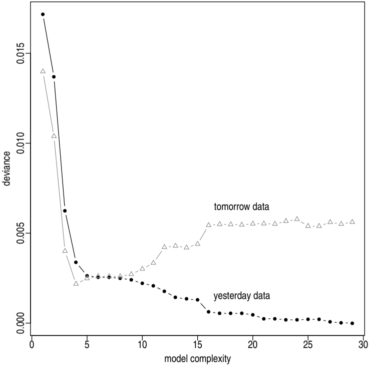
If we knew $f(x)$, we could plot separately bias and variance



Source: AS2012

A simple prototype problem

- But as we do not know $f(x)$, we only may compute the *residual variance* for the new (*tomorrow*) data:



This plot gives the residual deviance as function of the degree p, by using the model obtained with the *yesterday* data to predict the *tomorrow* data

A simple prototype problem

- When p (the model complexity indicator) increases, the fit improves on the *yesterday* data, but this is not true for the *tomorrow* data.
- goodness-of-fit measure is not a good indicator of the quality of the model
- When p increases too much, we 'overfit' the data and this indicates an excess of *optimism*!

- This happens because the model (the polynomial in the example) follows random fluctuations in yesterday's data not observed in the new sample (and not characteristic of the studied phenomenon), and it mistakes local (random) regularity with a systematic pattern.

A simple prototype problem

- So that... do not evaluate a model by using the same data used to fit it (the *yesterday* ones).
- If we want a more reliable evaluation, we need to use other data (the *tomorrow* ones)
- How?!

A simple prototype problem

- We need tools in order to select models:
  1. *training set* and *test (evaluation) set*
  2. cross-validation
  3. information criteria

Training set, test set

- If we have n data, and n is *large*, we can divide it in two groups randomly chosen:
  a training set used to fit the various candidate models and
  a test set (sometime called *evaluation set*) used to evaluate the performance of the available models and to choose the most accurate one.
- We compare results obtained with different models on the test set.
- This scheme reduces the sample size used for fitting the model, but this is not a problem when n is huge.
- Because the same test set can be used to evaluate many different models, there is a risk that the final assessment is still somewhat biased and too optimistic. Sometimes a third set of data, called validation set, is often created and used for final evaluation of the prediction error
- Examples of proportions for the sizes of the sets are:

training set	test set	validation set
50%	25%	25%
75%	25%	—

- training and test sets are somehow similar to what was done with *yesterday* and *tomorrow* data.

Information criteria

- The residual variance (or the deviance) is an unreliable indicator of the quality of the model, because it is too optimistic in evaluating the prediction error.
- We can penalize the *deviance* $D = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
- …or a monotonic transform:
  $-2\log L = n\,\log(D/n) + \text{(costant)}$
- with a suitable quantity quantifying the model complexity
- The $\log L$, for the gaussian model, has an interpretation as log-likelihood.
- Criteria that follow this logic can be traced back to objective functions such as

$$IC(p) = -2\log L + \text{penalty}(p)$$

- The choice of the specific penalty function identifies a particular criterion.

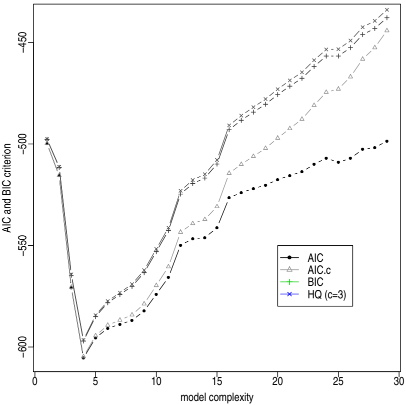
Information criteria

- Some possibile penalty are in the following table

criterion	author	penalty(p)
AIC	Akaike	$2p$
AIC$_c$	Sugiura, Hurvich-Tsay	$2p + \frac{2p\,(p+1)}{n-(p+1)}$
BIC/SIC	Akaike, Schwarz	$p\,\log n$
HQ	Hannan-Quinn	$c\,p\,\log\log n, \quad (c>2)$

- These criteria are applied also to *not nested* and *not gaussian* models.

Information criteria – example



We choose p minimising $IC(p)$ using some criteria in the previous table;
in our example all choices for penalty suggest $p = 4$.

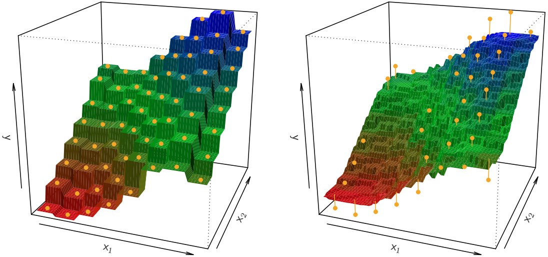
Non parametric regression

k-Nearest Neighbors: regression

Given a value k and a prediction point x_0, the KNN regression identifies in the training set the k nearest observations, N_0

$$\hat{f}(x_0) = \frac{1}{k} \sum_{x_i \in N_0} y_i$$

k-Nearest Neighbors: regression



KNN with $p = 2$, $k = 1$ (left) and $k = 9$ (right). With small k high variance and low bias, since prediction is performed on a single observation.

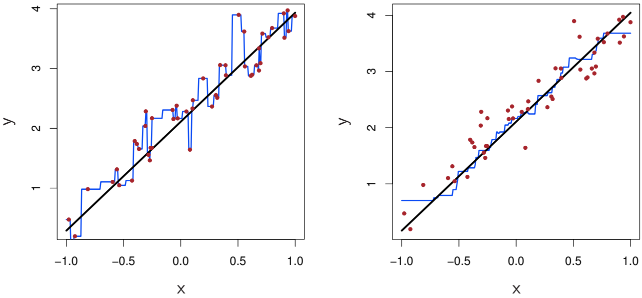
k-Nearest Neighbors: regression

The optimal value of k is related to the trade-off viarance-bias.

- small $k \rightarrow$ high variance and low bias
- big $k \rightarrow$ low variance (smoother prediction) and high bias -local structure of $f(X)$ may not be captured-

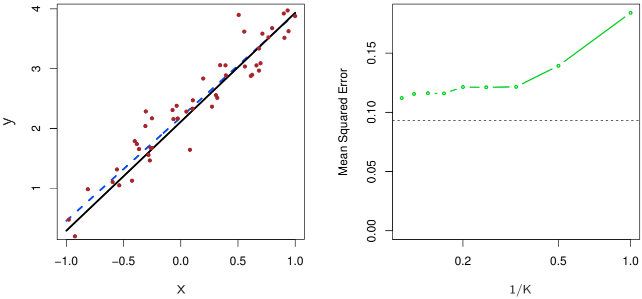
k-Nearest Neighbors: regression

Parametric approach may be preferred to the non parametric if the parametric form is close to the 'real' f.



Comparison between KNN with $k = 1$ (left) e $k = 9$ (right). Since the true relationship is linear the non parametric approach will have a worse performance.

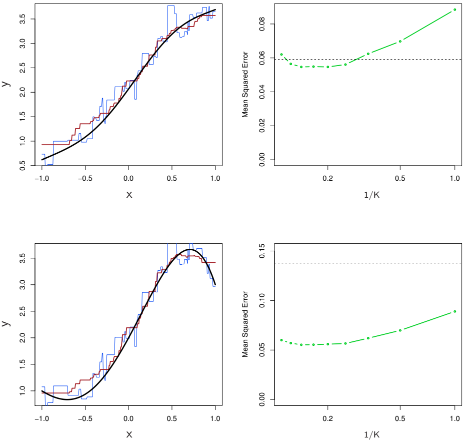
k-Nearest Neighbors: regression



Regression line (dashed line) Test MSE for regression line (dashed) and KNN (green) as function of $1/k$.

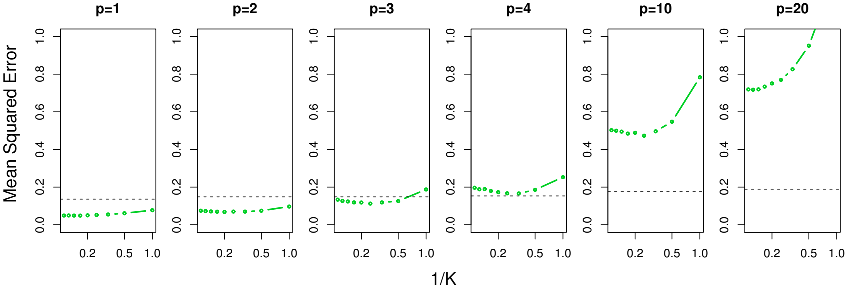
Best results for KNN are with high value of k.

k-Nearest Neighbors: regression



Nonlinear relationships and KNN with $k = 1$ (blue) and $k = 9$ (red). Conditional to nonlinearity of f the KNN performance changes with respect to LM. As the nonlinearity becomes more evident, the performance of KNN with high k will increase.

k-Nearest Neighbors: regression

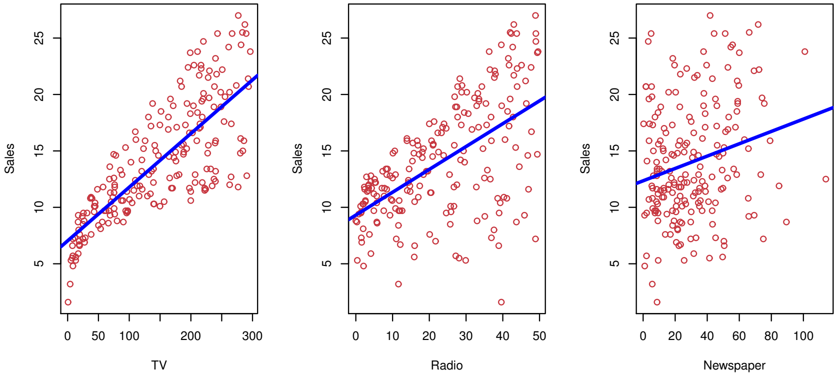


By increasing the number of variables p, the KNN performance will rapidly decrease in terms of MSE test.

It is more difficult to find the 'nearest neighbours' … curse of dimensionality

k-Nearest Neighbors: example

Sales of a product in thousands of units as function of budget in `tv`, `radio`, `newspapers` for 200 different markets.



Regression line for tv, radio, newspapers.

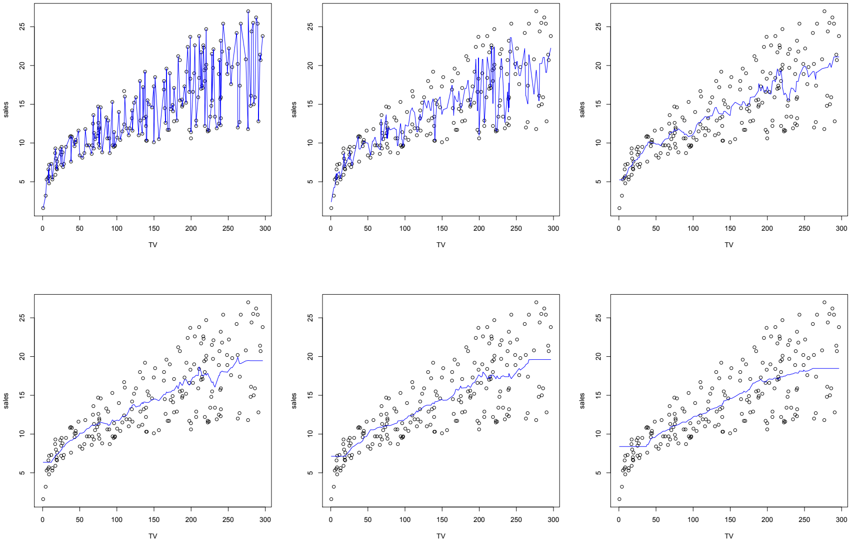
k-Nearest Neighbors: example

We wish to study the performance of KNN for some values of k with the only variable `tv`

k-Nearest Neighbors: example

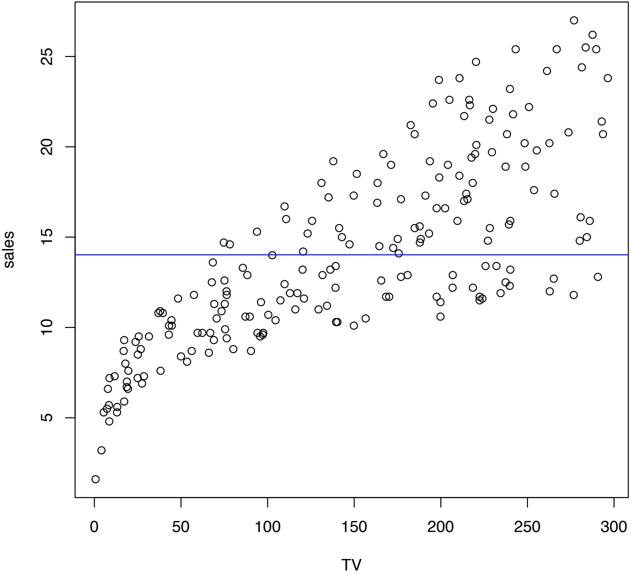
all data

k=1, 2, 10, 20, 30, 50

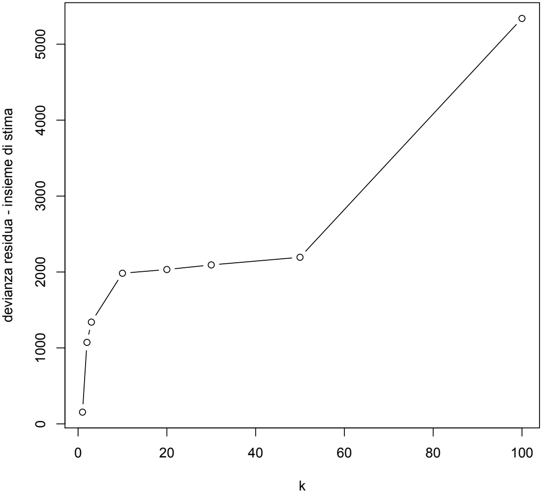


k-Nearest Neighbors: example

k=200



k-Nearest Neighbors: example

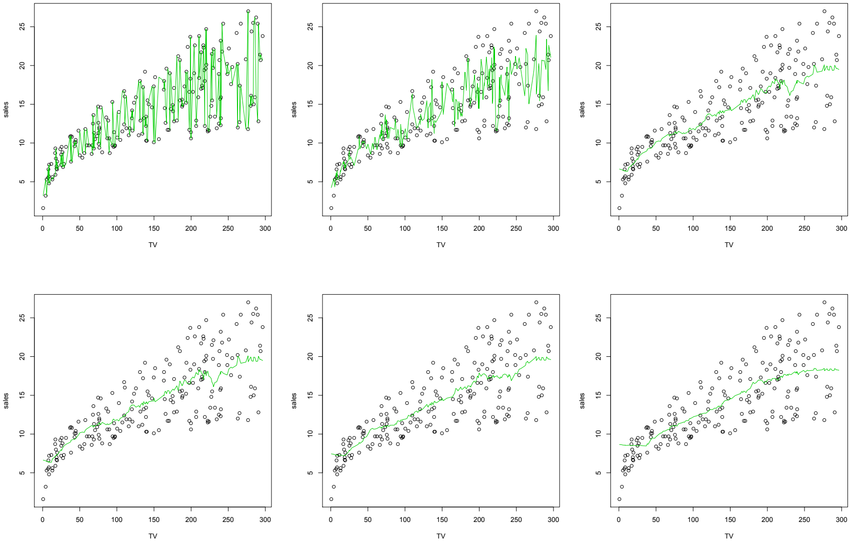


KNN performance decreases as k increases.

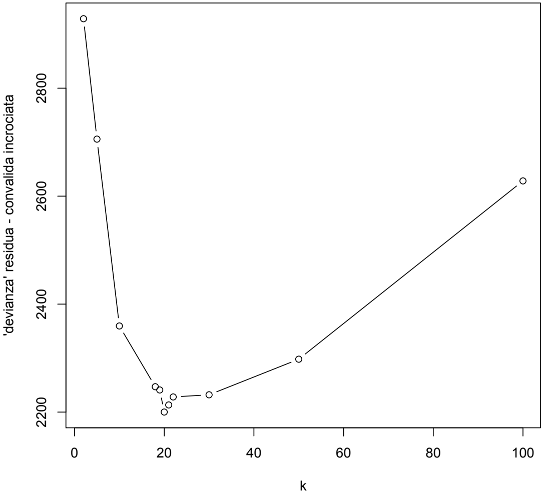
k-Nearest Neighbors: example

Cross validation leave-one-out

k=1, 2, 10, 20, 30, 50



k-Nearest Neighbors: example

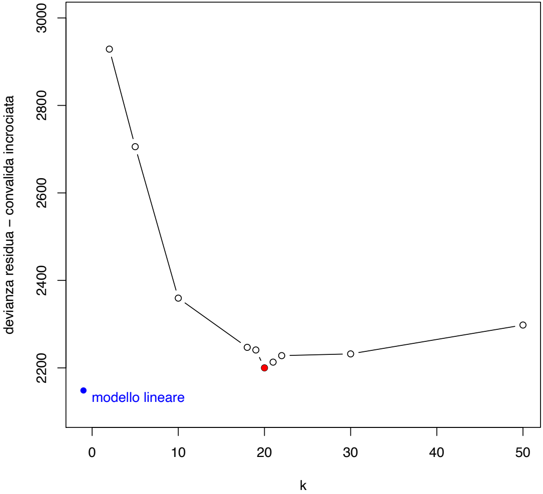


A minimum has reached . . .

trade-off between variance and bias

k-Nearest Neighbors: example

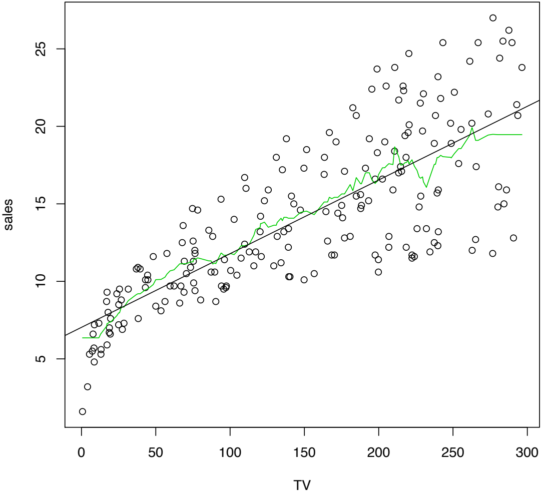
Performance of linear model and KNN with variable `tv`



In the case of `tv` the linear model performs better than the KNN for each value of k.

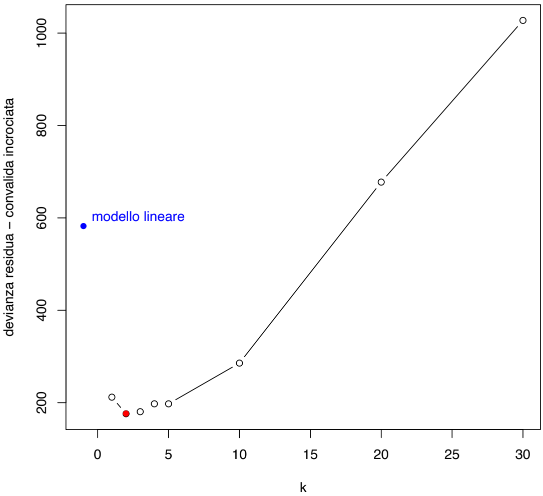
k-Nearest Neighbors: example

Linear model and KNN-20 with variable `tv`



k-Nearest Neighbors: example

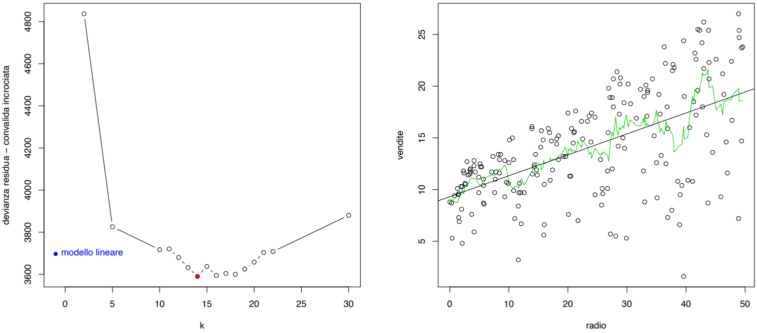
Performance of linear model and KNN with variable `tv` and `radio`



Adding the variable `radio` highly increases the performance of KNN. The minimum is reached for $k = 2$

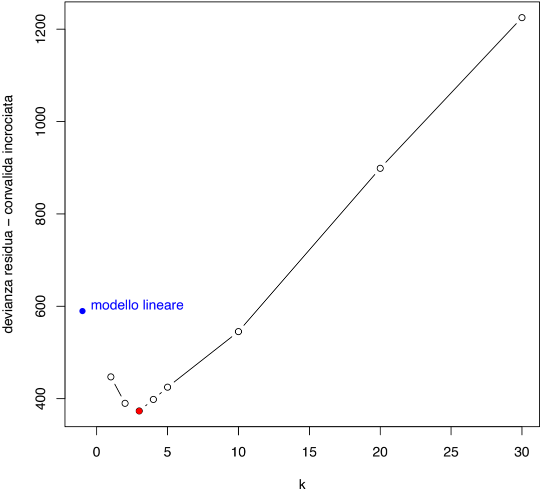
k-Nearest Neighbors: example

Performance of linear model and KNN with variable `radio`



k-Nearest Neighbors: example

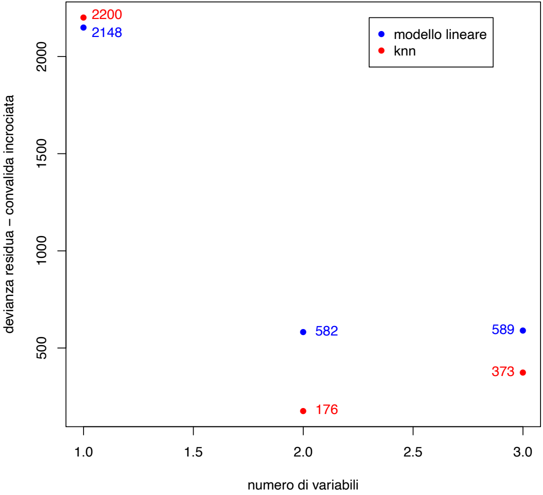
Let us add the variable `newspapers`



Adding the variable `newspapers` does not increase the performance of the model.

The KNN is better than the linear model in any case.

k-Nearest Neighbors: example



Local regression and `loess`

Local regression

- If $f(x)$ is a derivable function in x_0 then, the Taylor's approximation says that it is locally approximated by a line passing through $(x_0, f(x_0))$, i.e.,

$$f(x) = \underbrace{f(x_0)}_{\alpha} + \underbrace{f'(x_0)}_{\beta}(x - x_0) + \text{error}$$

- We introduce the weighted least squares by weighting observations x_i with their distance from x_0:

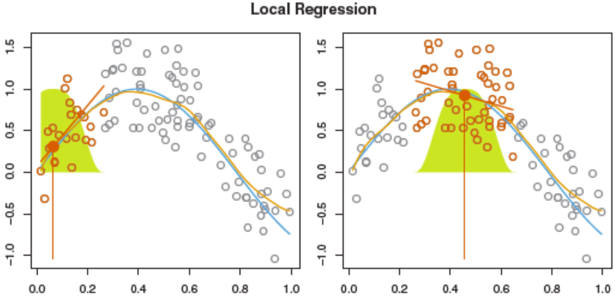
$$\min_{\alpha,\beta} \sum_{i=1}^{n} \left\{ y_i - \alpha - \beta(x_i - x_0) \right\}^2 w_h(x_i - x_0)$$

  - h ($h > 0$) is a scale factor, called bandwidth or smoothing parameter, and
  - $w_h(\cdot)$ is a symmetric density function around 0, said kernel.

Local regression

- By varying x_0, we obtain a whole estimated curve $\hat{f}(x)$.
- The most important component is h, which regulates the smoothness of the curve, while the choice of w is less relevant.
- We could think to w as the density of the normal distribution $N(0,1)$

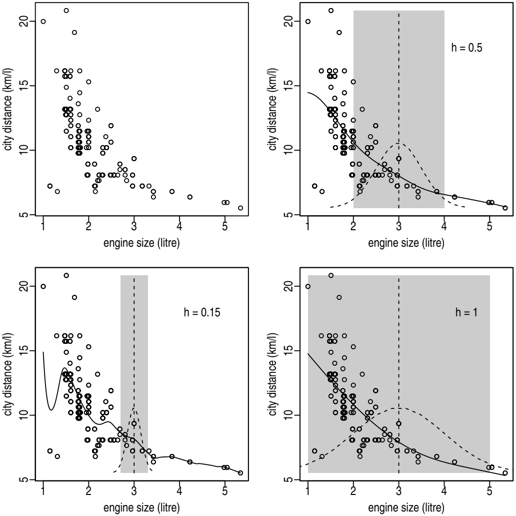
Local regression



Local regression: blue curve represents the real $f(x)$, light orange curve corresponds to the local regression estimate $\hat{f}(x)$. The orange colored points are local to the target point x_0, represented by the orange vertical line. The yellow bell-shape superimposed on the plot indicates weights assigned to each point, decreasing to zero with distance from the target point. The fit $\hat{f}(x)$ at x_0 is obtained by fitting a weighted linear regression (orange line segment), and using the fitted value at x_0 (orange solid dot) as the estimate $\hat{f}(x_0)$

Local regression

The effect of h is relevant



Variable bandwidths and `loess`

- in many cases, there is an advantage in using a non constant bandwidth along the x-axis, according it to the level of sparseness of observed points
- variable bandwidth: it is reasonable to use larger values of h when x_i are more scattered
- Good idea! . . . but how do we modify h?
- loess: express the smoothing parameter defining the fraction of effective observations for estimating $f(x)$ at a certain point x_0 on the x-axis;
- this fraction is kept constant
- this implies automatically a setting of the bandwidth related to the sparsity of data