



Inteligencia Artificial

UTN – FRVM

5º Año Ing. en Sistemas de
Información



Agenda

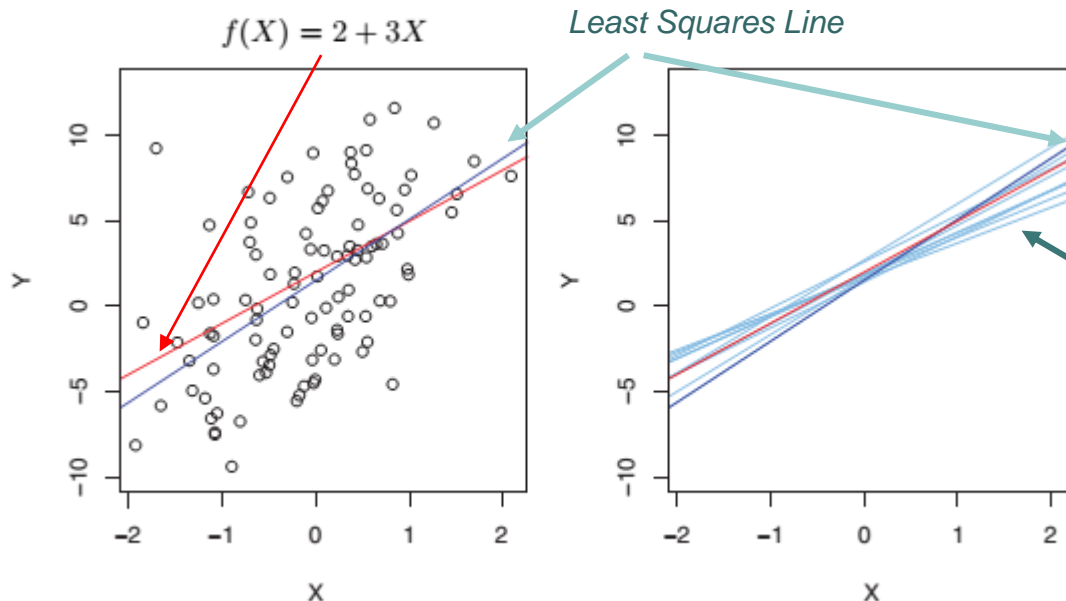


- *Accuracy of the Coefficient Estimates.*
- K-Nearest Neighbours.
- Resampling Methods.

Accuracy of the Coefficient Estimates

$$Y = \beta_0 + \beta_1 X + \epsilon.$$

- Here β_0 is the intercept term—that is, the expected value of Y when $X = 0$, and β_1 is the slope—the average increase in Y associated with a one-unit increase in X .
- We typically assume that the error term is independent of X .



Separate random set of observations. Each least squares line is different, but on average, the least squares lines are quite close to the population regression line.

Standard Error

- The average of $\hat{\mu}$'s over many data sets will be very close to μ .
- How far off will that single estimate of $\hat{\mu}$ be?

$$\text{Var}(\hat{\mu}) = \text{SE}(\hat{\mu})^2 = \frac{\sigma^2}{n}$$

Standard deviation of each realization of y_i of Y

the standard error tells us the average amount that this estimate $\hat{\mu}$ differs from the actual value of μ

$$\text{SE}(\hat{\beta}_0)^2 = \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right], \quad \text{SE}(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad \sigma^2 = \text{Var}(\epsilon)$$



Standard Error

- In general, σ^2 is not known, but can be estimated from the data.
- The estimate of σ is known as the *residual standard error (RSE)*

Residual sum of squares

$$\text{RSE} = \sqrt{\text{RSS}/(n-2)}.$$


$$\text{RSS} = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + (y_2 - \hat{\beta}_0 - \hat{\beta}_1 x_2)^2 + \dots + (y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n)^2$$

Standard errors can be used to compute *confidence intervals*. A 95 % confidence interval is defined as a range of values such that with 95 % interval probability, the range will contain the true unknown value of the parameter.

$$\hat{\beta}_1 \pm 2 \cdot \text{SE}(\hat{\beta}_1)$$

Hypothesis tests

- H_0 : There is no relationship between X and Y
(Null Hypothesis) $\beta_1 = 0 \quad Y = \beta_0 + \epsilon$
- H_a : There is some relationship between X and Y
(Alternative Hypothesis)

$$t = \frac{\hat{\beta}_1 - 0}{SE(\hat{\beta}_1)}$$

	Coefficient	Std. error	t-statistic
Intercept	7.0325	0.4578	15.36
TV	0.0475	0.0027	17.67

Assessing the Accuracy of the Model

- The quality of a linear regression fit is typically assessed using two related quantities: the *residual standard error* (RSE) and the R^2 statistic.
- **RSE**: it is the average amount that the response will deviate from the true regression line. Problem? Units of Y (Which is a good RSE?)

$$\text{RSE} = \sqrt{\frac{1}{n-2} \text{RSS}} = \sqrt{\frac{1}{n-2} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

- R^2 : takes the form of a *proportion* —the *proportion of variance explained*— and so it always takes on a value between 0 and 1, and is independent of the scale of Y

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}$$

the amount of variability that is left unexplained after performing the regression

$$\text{TSS} = \sum (y_i - \bar{y})^2$$

the amount of variability in the response that is removed by performing the regression.

the amount of variability inherent in the response before the regression is performed

K-Nearest Neighbours

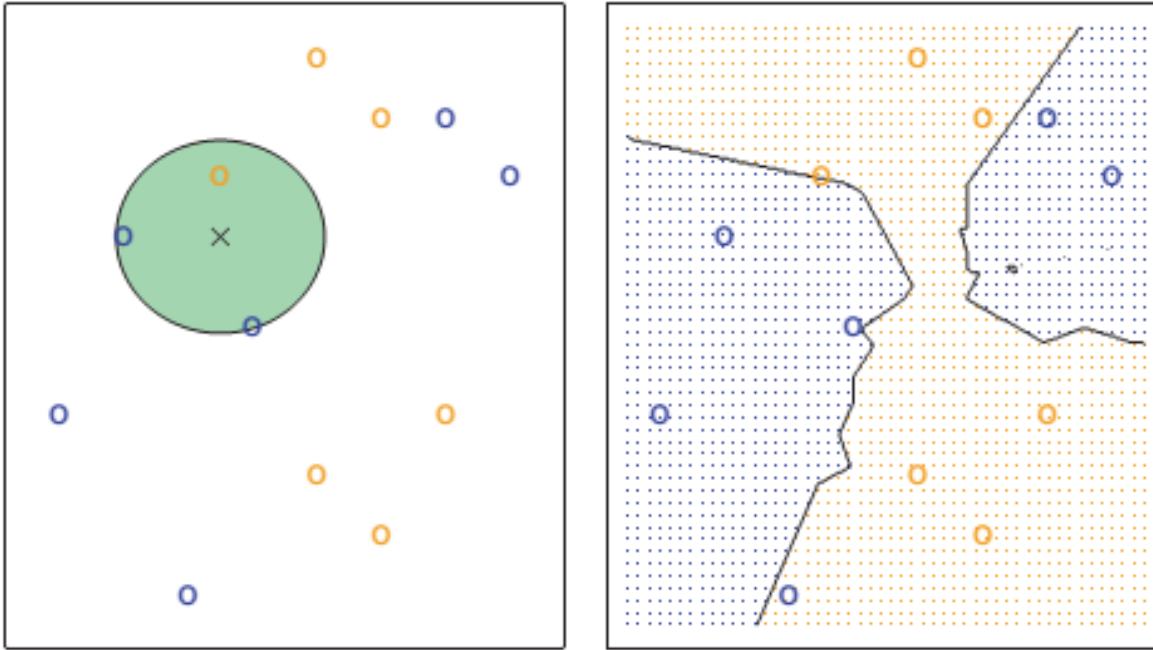
- Many approaches attempt to estimate the *conditional distribution* of Y given X , and then classify a given observation to the class with highest *estimated* probability.
- Given a positive integer K and a test observation x_0 , the *KNN* classifier first identifies the neighbors K points in the training data that are closest to x_0 , represented by N_0
- The conditional probability for class j as the fraction of points in N_0 whose response values equal j :

$$\Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j)$$

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^p (x_{ri} - x_{rj})^2}$$

$$d = \|\mathbf{x} - \mathbf{y}\| = \sqrt{\sum_{i=1}^n |x_i - y_i|^2}.$$

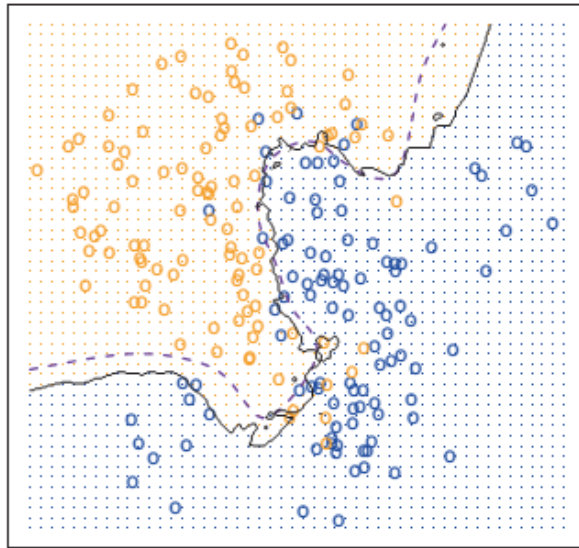
KNN



- The **KNN** approach, using $K = 3$, is illustrated in a simple situation with six blue observations and six orange observations.
- **Left:** a test observation at which a predicted class label is desired is shown as a black cross. The three closest points to the test observation are identified, and it is predicted that the test observation belongs to the most commonly-occurring class, in this case blue.
- **Right:** The KNN decision boundary for this example 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.

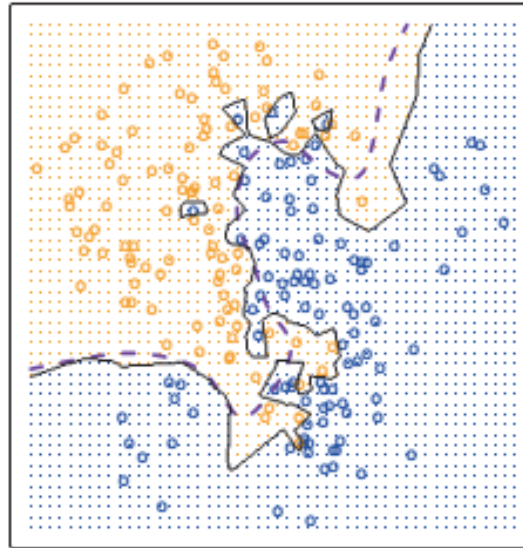
KNN

KNN: K=10



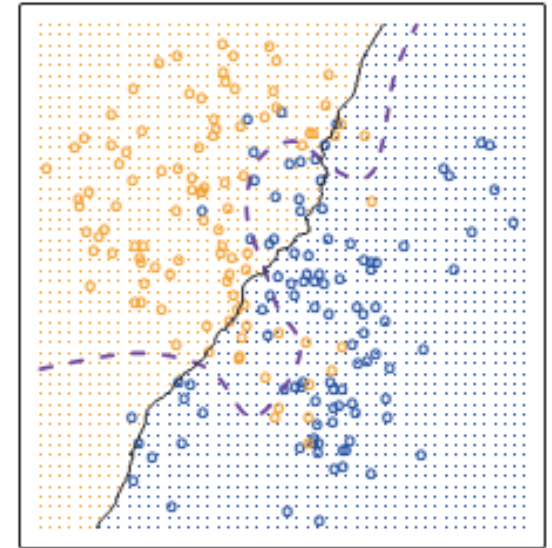
x_1

KNN: K=1



0.1695

KNN: K=100

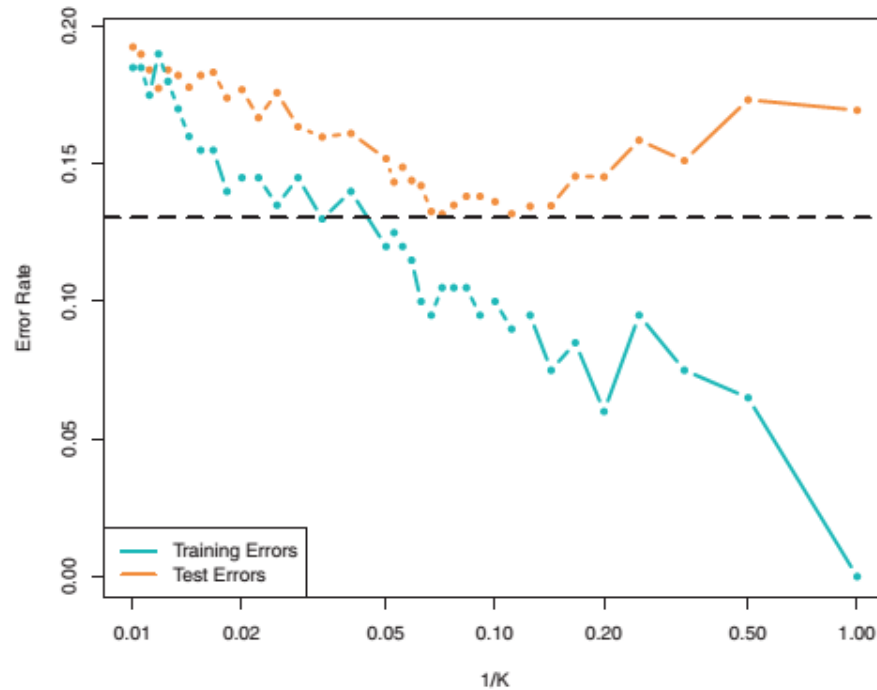


0.1925

Test Errors 0.1363

- The black curve indicates the KNN decision boundary using $K = 10, 1, 100$. The Bayes decision boundary is shown as a purple dashed line.

KNN Training-Test Errors



- The **KNN** training error rate (blue, 200 observations) and test error rate (orange, 5,000 observations), as the level of flexibility (assessed using $1/K$) increases, or equivalently as the number of neighbors K decreases. The black dashed line indicates the Bayes error rate. The jumpiness of the curves is due to the small size of the training data set.



Resampling Methods

- **Resampling methods** involve repeatedly drawing samples from a training set and refitting a model of interest on each sample in order to obtain additional information about the fitted model.
- In order to estimate the variability of a linear regression fit, we can repeatedly draw different samples from the training data, fit a linear regression to each new sample, and then examine the extent to which the resulting fits differ.
- Such an approach may allow us to obtain information that would not be available from fitting the model only once using the original training sample.
- Resampling approaches can be computationally expensive, because they involve fitting the same statistical method multiple times using different subsets of the training data.
- **Cross-validation** and the **bootstrap**.
- **Cross-validation** can be used to estimate the test error associated with a given statistical learning method in order to evaluate its performance, or to select the appropriate level of flexibility.
- The process of evaluating a model's performance is known as *model assessment*, whereas the process of selecting the proper level of flexibility for a model is known as *model selection*.
- The **bootstrap** is used in several contexts, most commonly to provide a measure of accuracy of a parameter estimate or of a given selection statistical learning method.

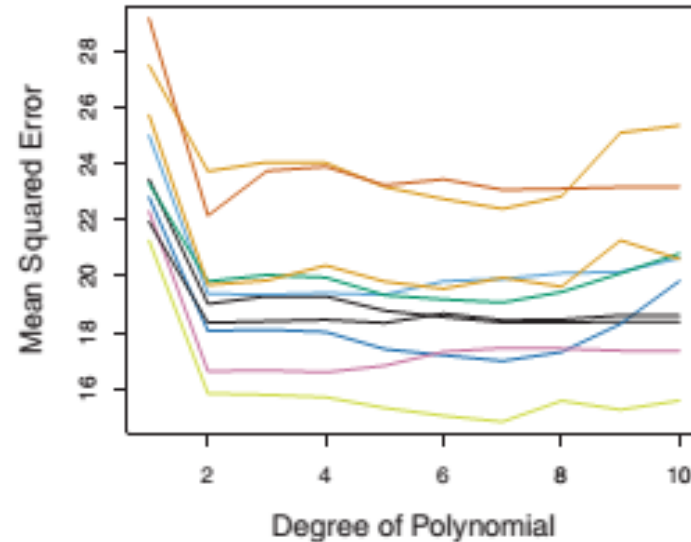
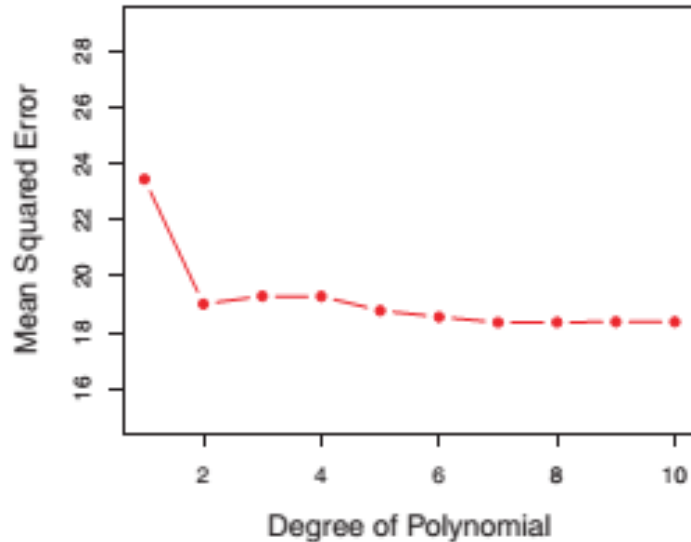
Cross-Validation

- In the absence of a very large designated test set that can be used to directly estimate the test error rate, a number of techniques can be used to estimate this quantity using the available training data.
- *The Validation Set Approach*



- *A schematic display of the validation set approach. A set of n observations are randomly split into a training set (shown in blue, containing observations 7, 22, and 13, among others) and a validation set (shown in beige, and containing observation 91, among others). The statistical learning method is fit on the training set, and its performance is evaluated on the validation set.*

Validation set approach



- The validation set approach was used on the Auto data set in order to estimate the test error that results from predicting mpg using polynomial functions of horsepower.
- Left: Validation error estimates for a single split into training and validation data sets.
- Right: The validation method was repeated ten times, each time using a different random split of the observations into a training set and a validation set. This illustrates the variability in the estimated test MSE that results from this approach.

Leave - One - Out Cross - Validation

- Involves splitting the set of observations into two parts.
- However, instead of creating two subsets of comparable size, a single observation (x_1, y_1) is used for the validation set, and the remaining observations $\{(x_2, y_2), \dots, (x_n, y_n)\}$ make up the training set.

1 2 3 n



1 2 3 n

1 2 3 n

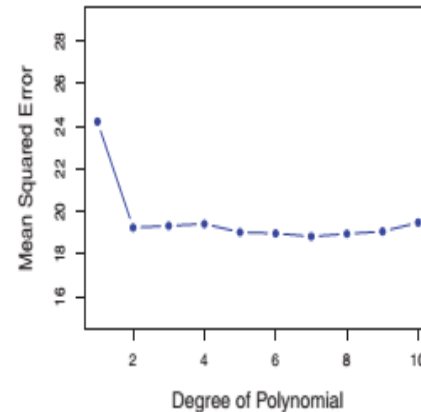
1 2 3 n

...

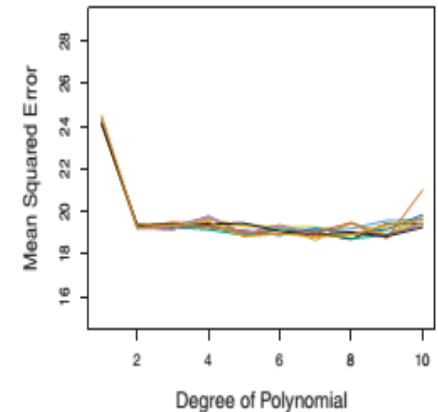
1 2 3 n

$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^n \text{MSE}_i.$$

LOOCV



10-fold CV





k-Fold Cross-Validation

- This approach involves randomly dividing the set of observations into k groups, or *folds*, of approximately equal size.
- The first fold is treated as a validation set, and the method is fit on the remaining $k - 1$ folds.
- The mean squared error, MSE_1 , is then computed on the observations in the held-out fold. This procedure is repeated k times



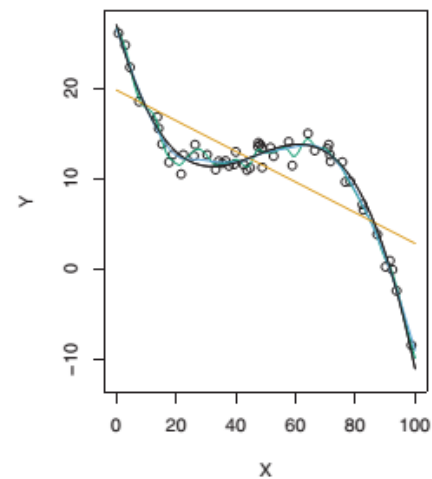
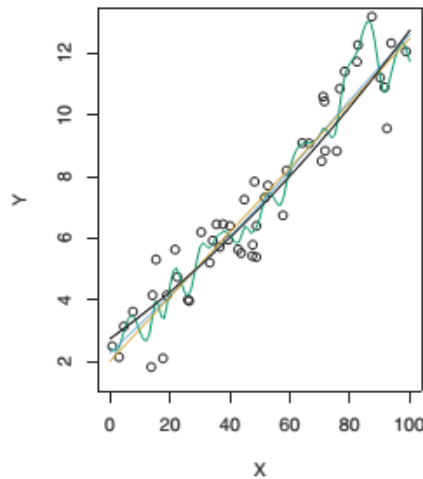
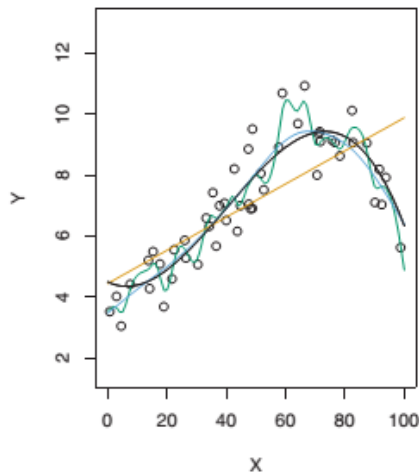
k-Fold Cross-Validation

- This process results in k estimates of the test error, $\text{MSE}_1, \text{MSE}_2, \dots, \text{MSE}_k$. The k -fold CV estimate is computed by averaging these values.

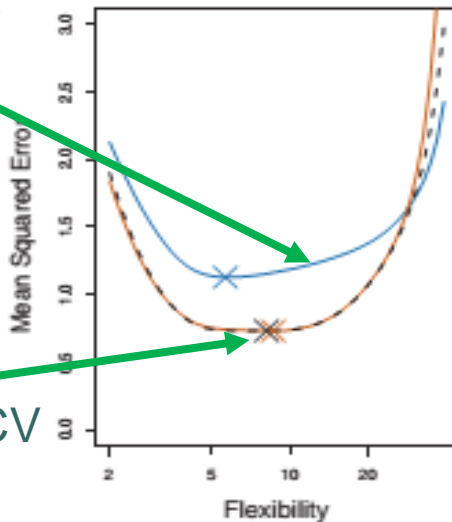
$$\text{CV}_{(k)} = \frac{1}{k} \sum_{i=1}^k \text{MSE}_i.$$

- Cross-validation is a very general approach that can be applied to almost any statistical learning method.

k-Fold Cross-Validation



LOOCV



10-fold CV

