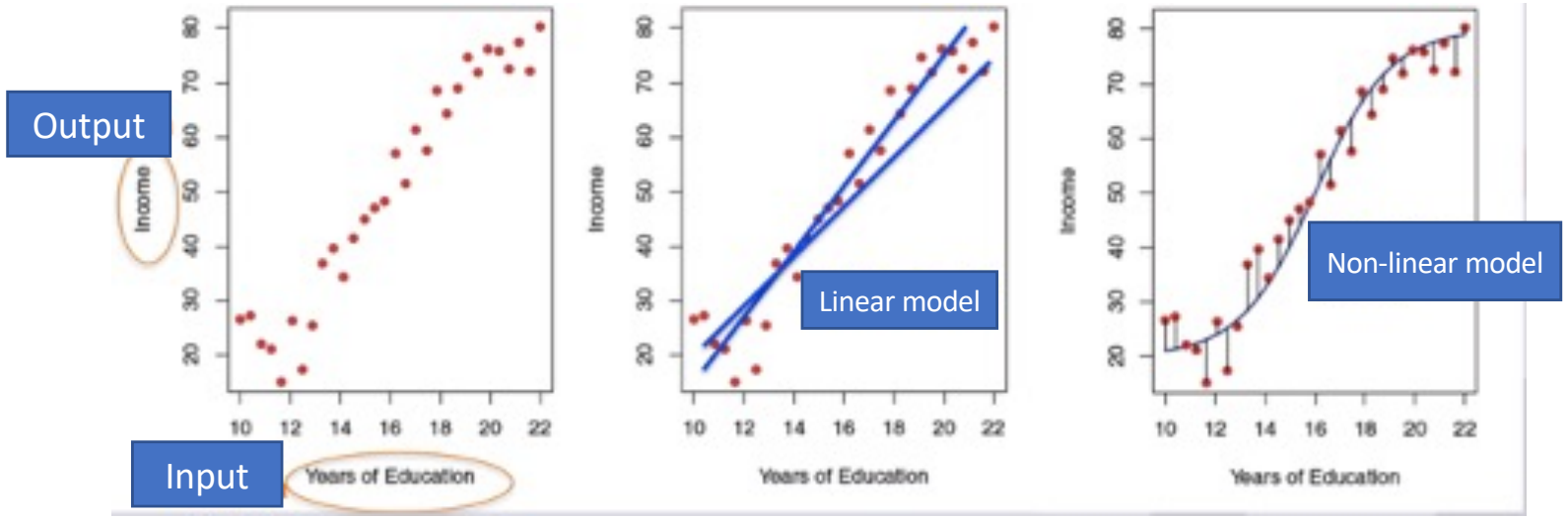
Unit 2.
Parametric regression

Artificial Intelligence and Learning

Contents

2.1 Introduction



Example obtained from “An introduction to statitistical learning with applications in R”. Autores: G. James, D. Witten, T. Hastie, R. Tibshirani. Editorial: Springer 2013.

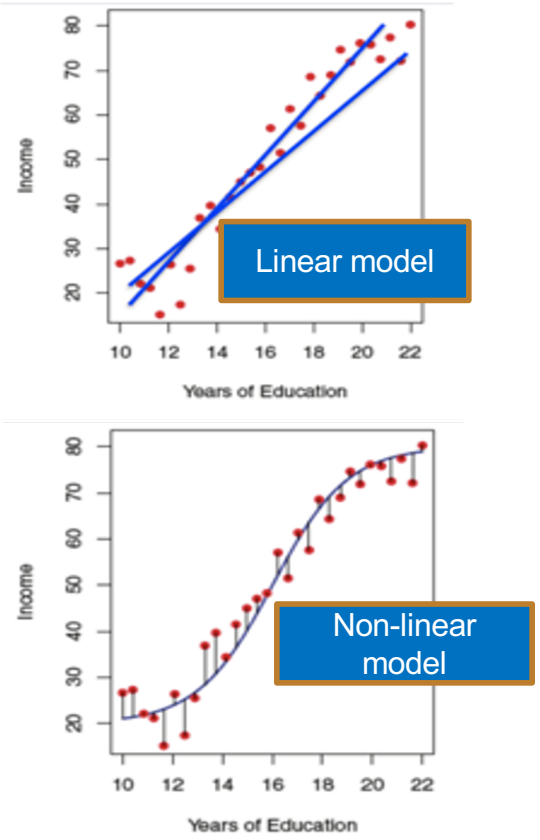
For the same data set, it is possible to design several models. In this scenario, how do you choose the most appropriate model for a data set?

2.1 Introduction. Cost function

We can evaluate a certain function that depends on the error (called **cost function**) and choose the model that provides the **lowest cost**.

The cost function will depend on the estimation errors, also called residuals.

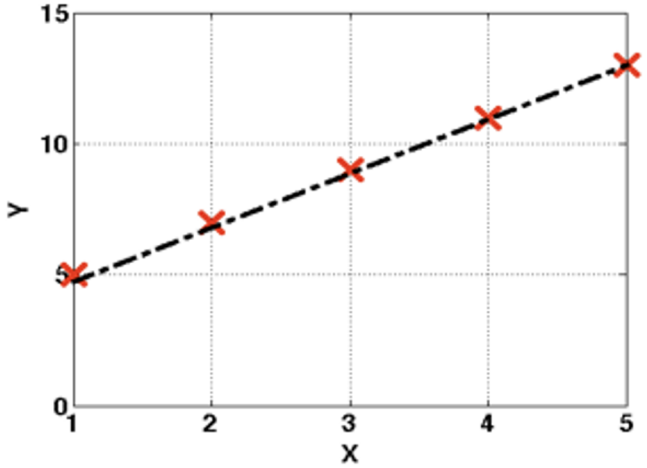
The residuals are obtained as the **difference between the real value and the value estimated** by the model (vertical lines in the figure on the right).



2.1 Introduction. Cost function

Intuitively, if we have T pairs of points (X, Y), we can see if they are distributed linearly:

$$5 = a + b * 1$$
$$7 = a + b * 2$$
$$9 = a + b * 3$$
$$11 = a + b * 4$$
$$13 = a + b * 5$$



What is the value of a and b?

a is the intercept
b is the slope of the line

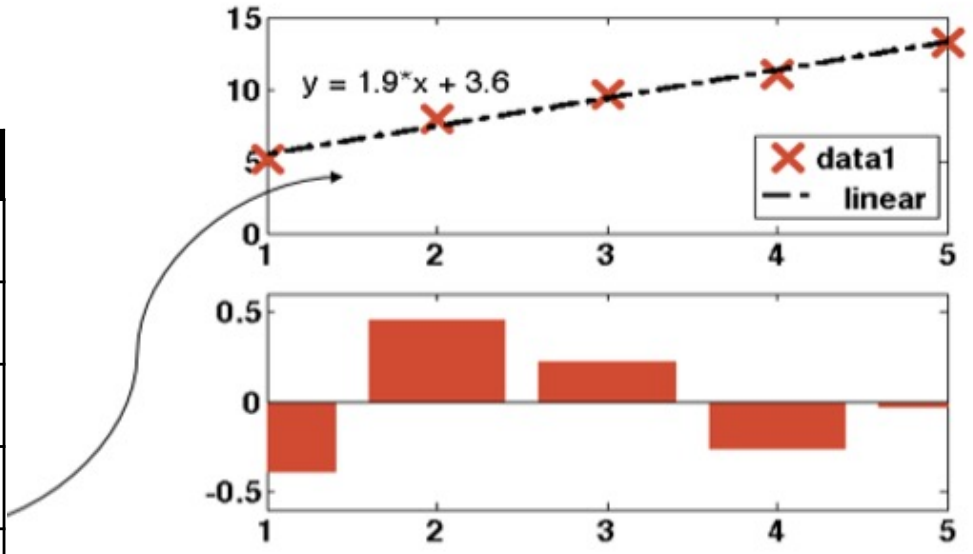
$f(x)=3+2*x$

a=3; b=2

2.1 Introduction. Cost function

Intuitively, if we have T pairs of points (X, Y), we can see if they are distributed linearly. Now, we have the following real values:

x	y
1	5.12
2	7.90
3	9.63
4	11.09
5	13.27



2.1 Introduction. Cost function

Estimated values

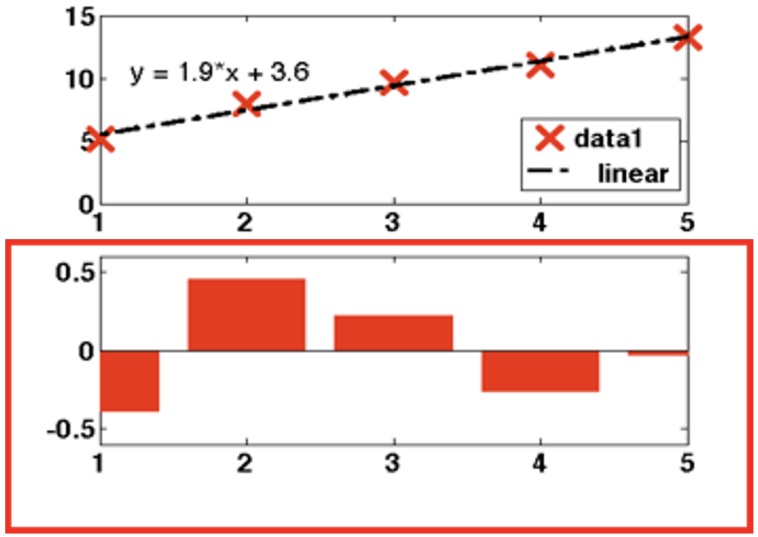
$$\hat{y} = 3.6 + 1.9 * x$$

5.5. =3.6 + 1.9 * 1
7.4. =3.6 + 1.9 * 2
9.4 =3.6 + 1.9 * 3
11.2=3.6 + 1.9 * 4
13.1=3.6 + 1.9 * 5

Real values

x	y
1	5.12
2	7.90
3	9.63
4	11.09
5	13.27

x	Residual
1	5.12 - 5.5. = -0.38
2	7.90 - 7.4. = 0.44
3	11.2 - 11.09 = 0,23
4	9.63 - 9.3 = -0.26
5	13.27 - 13.1 = -0.03



Residual = $\mathbf{y} - \hat{\mathbf{y}}$

Residuals are obtained as the **difference between the real value and the value estimated** by the model

2.1 Introduction. Cost function

In a general way:

$$f(\mathbf{x}) = \hat{\mathbf{y}} = w_0 + w_1\mathbf{x}$$

where w_i are the parameters of the model

How are they chosen?

Idea: choose w_i such that the distance between f(x) and $\mathbf{y}$ is minimal for **training data.**

Least squares: minimizing a cost function. In this case, the cost/objective function is the sum of squared called, **least squares:**

$$\min_{w_0, w_1} (f(\mathbf{x}) - \mathbf{y})^2$$

2.1 Introduction. Cost function

Idea: choose w_i such that the difference between f(x) and $\mathbf{y}$ is minimal for training data.

$$\min_{w_0, w_1} (f(\mathbf{x}) - \mathbf{y})^2$$

We minimize on the training set values:

Cost function:

$$\frac{1}{2m} \sum_{i=1}^{m} (\underbrace{f(x^{(i)})}_{} - \underbrace{y^{(i)}}_{})^2$$

$f(x^{(i)}) = w_0 + w_1 x^{(i)}$ Real values

Note: 2m is a constant that helps cancel 2 in derivative of the function when doing calculations for gradient descent

2.1 Introduction. Cost function

How to minimize the cost/objective function?

- **Gradient descent:** consider the derivative w.r.t the coefficients:

    * Batch gradient descent

    * Stochastic gradient descent

- **Close form solution:** compute gradient and set the gradient to zero, solving in closed form.

2.1 Introduction. Cost function

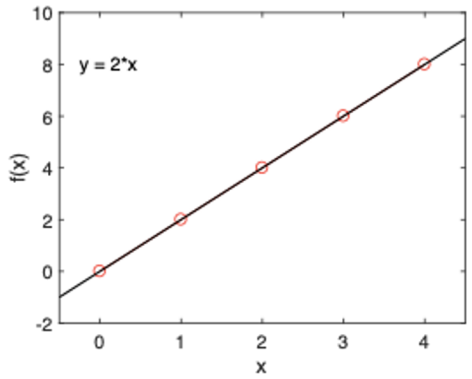
Cost function: $$J(w_0, w_1) = \frac{1}{2m} \sum_{i=1}^{m} (f(\mathbf{x}^{(i)}) - \mathbf{y}^{(i)})^2$$

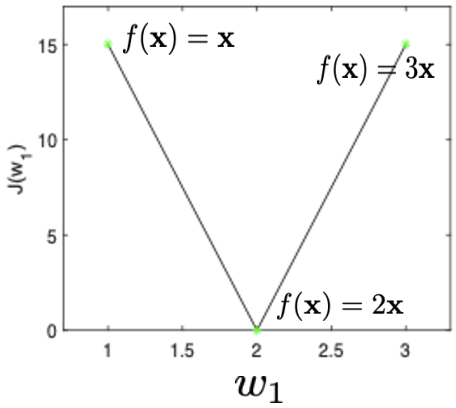
Hypothesis: $$f(\mathbf{x}) = \hat{\mathbf{y}} = w_0 + w_1 \mathbf{x}$$

Parameters: $$w_0, w_1$$

Objective: $$\min_{w_0, w_1} J(w_0, w_1)$$

2.1 Introduction. Cost function





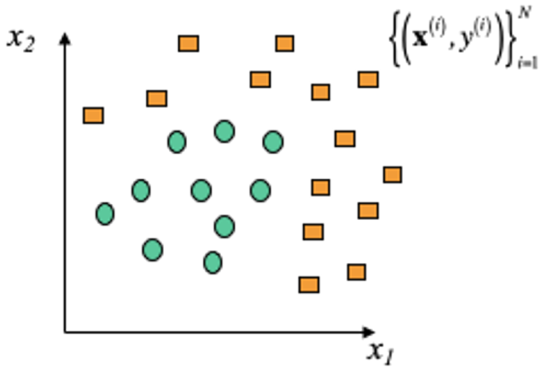
If we set $w_0 = 0 \rightarrow f(\mathbf{x}) = w_1\mathbf{x}$

We can intuitively see what the optimal value of w_1

		$f(\mathbf{x}) = w_1\mathbf{x}$		
x	y	$w_1 = 1$	$w_1 = 2$	$w_1 = 3$
0	0	0	0	0
1	2	1	2	3
2	4	2	4	6
3	6	3	6	9
4	8	4	8	12
$J(w_1)$		15	0	15

$$J(w_1) = \frac{1}{2m}\sum_{i=1}^{5}(f(x)^{(i)} - y^{(i)})^2$$

2.1 Introduction. Generalization and overfitting

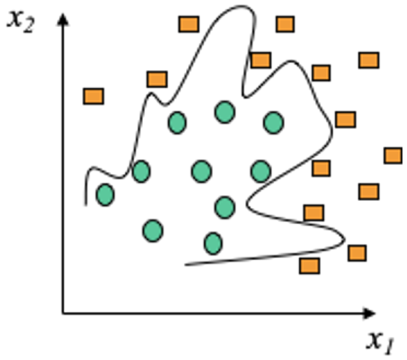


$$f_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2 + w_5 x_1^3 + \ldots$$

Cost function to be optimized. Mean square error

$$\frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - f_{\mathbf{w}}(\mathbf{x}^{(i)})\right)^2$$

2.1 Introduction. Generalization and overfitting



$$\left\{\left(\mathbf{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N}$$

One possible solution could be the borderline illustrated in this figure

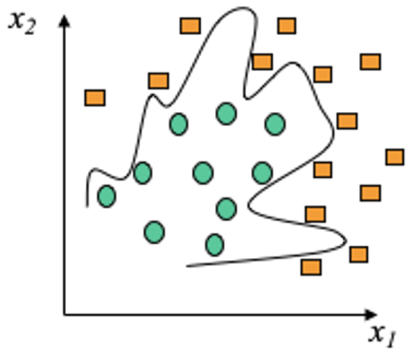
Complex

$$f_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2 + w_5 x_1^3 + \ldots$$

Cost function to be optimized. Mean square error

$$\frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - f_{\mathbf{w}}(\mathbf{x}^{(i)})\right)^2$$

2.1 Introduction. Generalization and overfitting



(*Overfitting*)

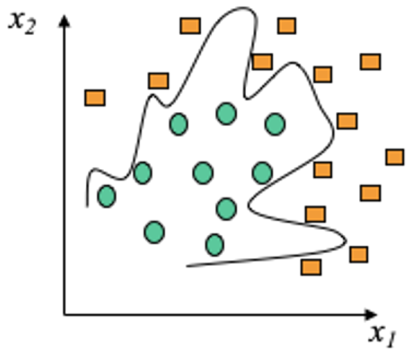
Cost function to be optimized. Mean square error

$$\frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - f_{\mathbf{w}}(\mathbf{x}^{(i)})\right)^2$$

Cost function to be optimized. Mean square error with regularization

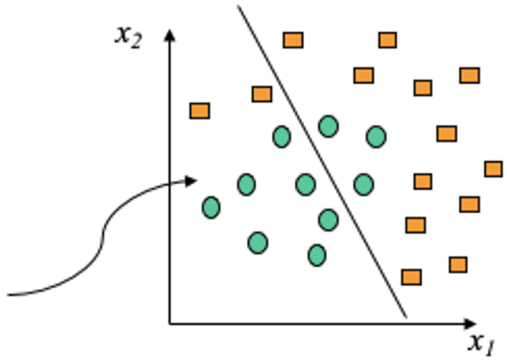
$$\frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - f_{\mathbf{w}}(\mathbf{x}^{(i)})\right)^2 + \lambda C_{\mathbf{w}}^{regulariz}$$

2.1 Introduction. Generalization and overfitting



(Overfitting)

The value of lambda is high, the learning algorithm will pay more attention to building a smooth boundary than to minimizing the difference between the model output and the desired value



(Underfitting)

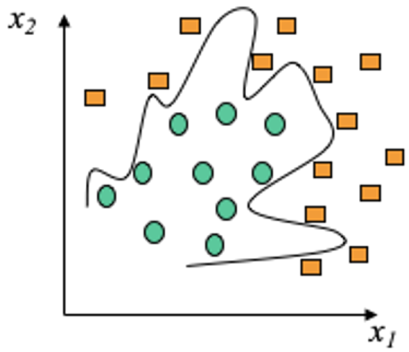
Cost function to be optimized. Mean square error

$$\frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - f_{\mathbf{w}}(\mathbf{x}^{(i)})\right)^2$$

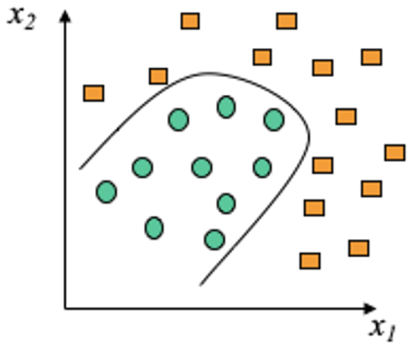
Cost function to be optimized. Mean square error with regularization

$$\frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - f_{\mathbf{w}}(\mathbf{x}^{(i)})\right)^2 + \lambda C_{\mathbf{w}}^{regulariz}$$

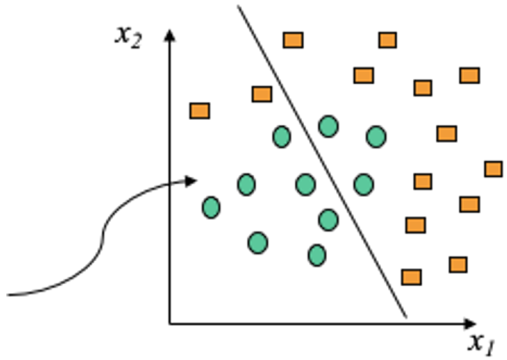
2.1 Introduction. Generalization and overfitting



(*Overfitting*)



(Good Generalization)



(*Underfitting*)

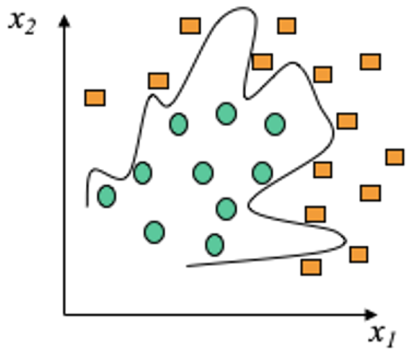
Cost function to be optimized. Mean square error

$$\frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - f_{\mathbf{w}}(\mathbf{x}^{(i)})\right)^2$$

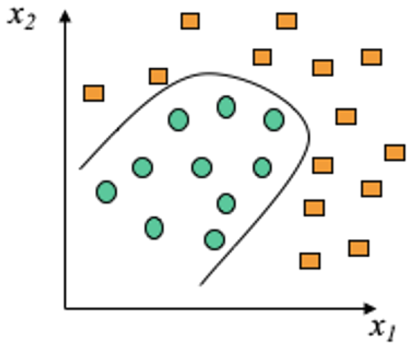
Cost function to be optimized. Mean square error with regularization

$$\frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - f_{\mathbf{w}}(\mathbf{x}^{(i)})\right)^2 + \lambda C_{\mathbf{w}}^{regulariz}$$

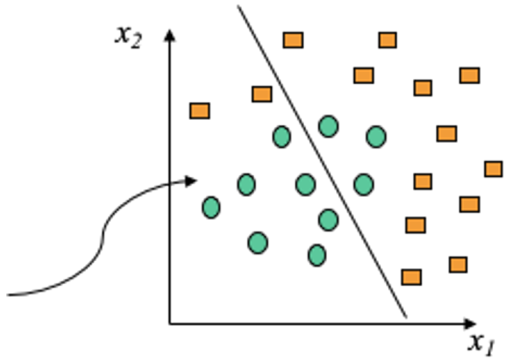
2.1 Introduction. Generalization and overfitting



(Overfitting)



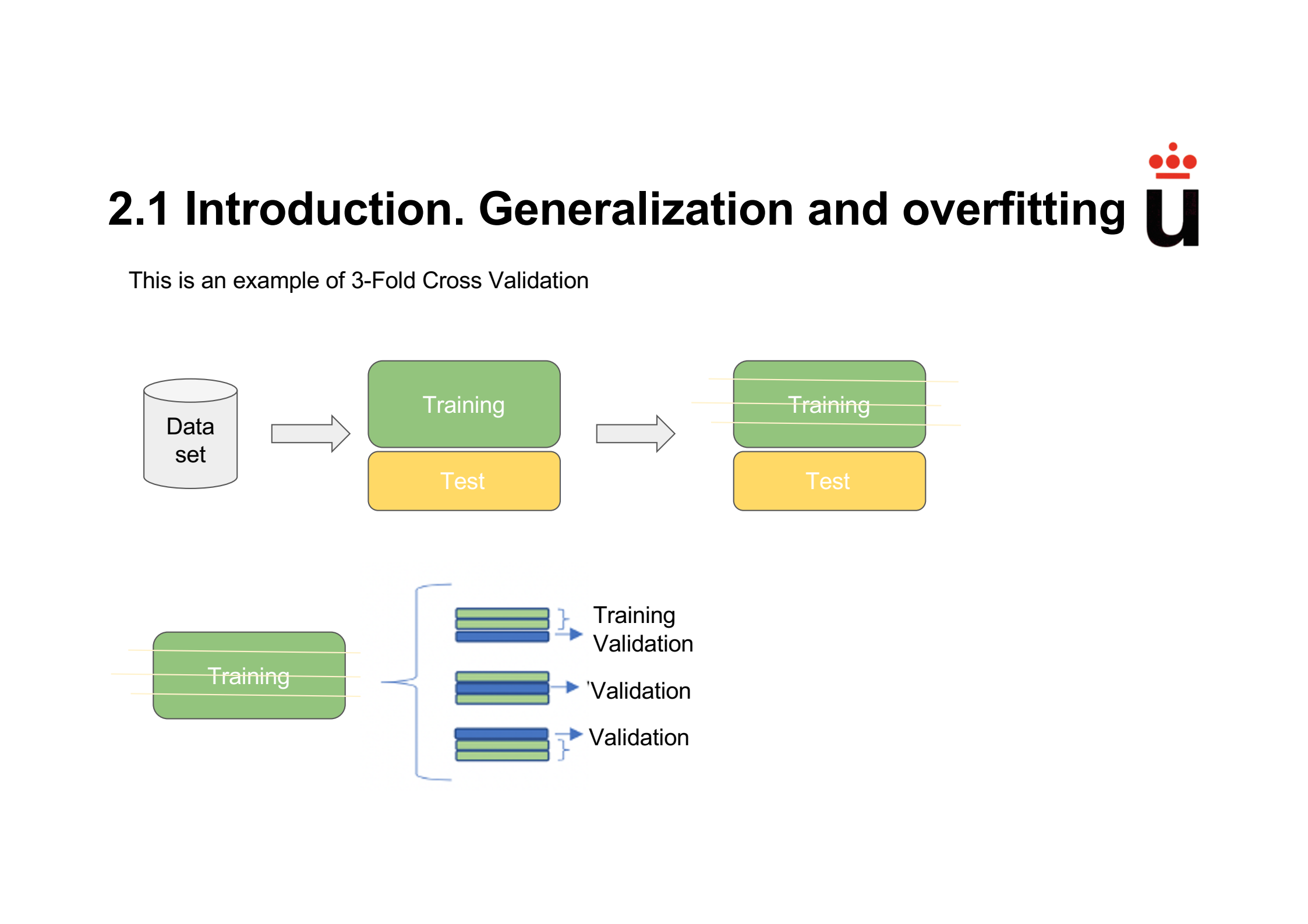
(Good Generalization)



(Underfitting)

To achieve a model with good GENERALIZATION capacity, it is possible to evaluate the performance of the model with **cross validation techniques**

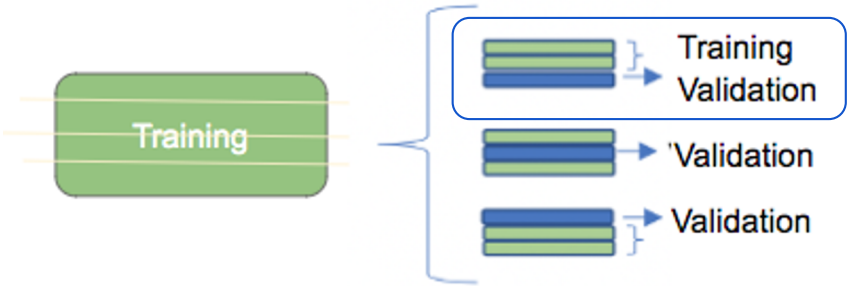
2.1 Introduction. Generalization and overfitting



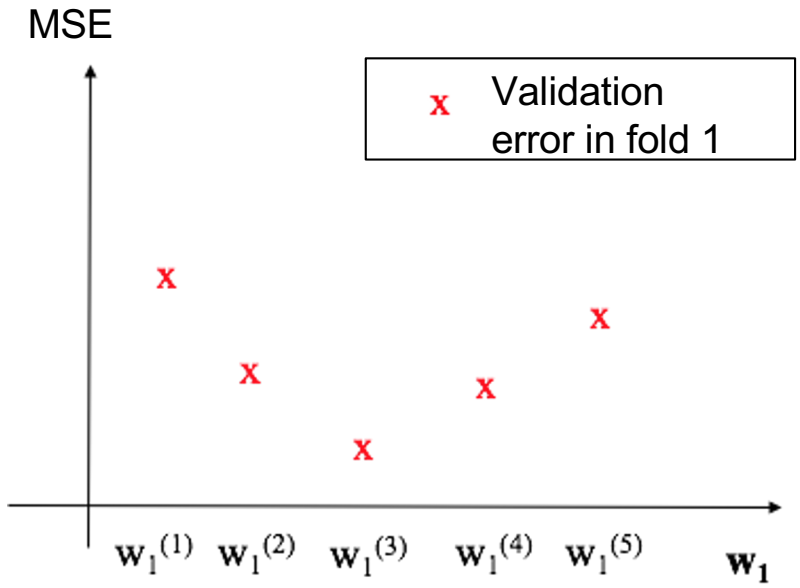
This is an example of 3-Fold Cross Validation

2.1 Introduction. Generalization and overfitting

This is an example of 3-Fold Cross Validation

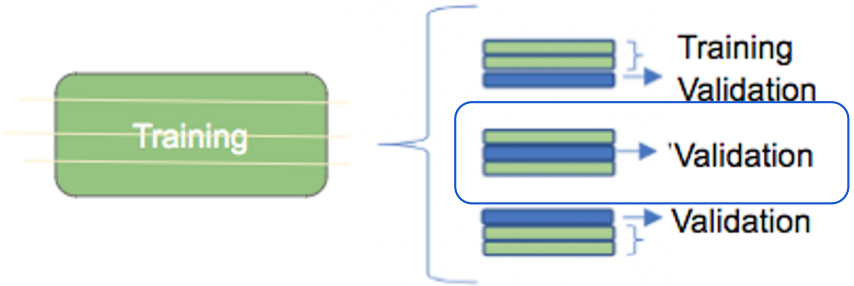


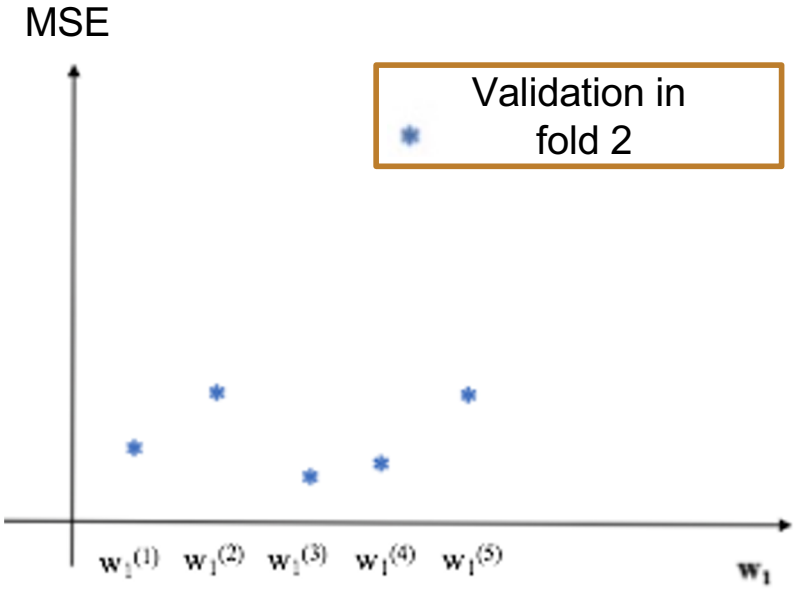
$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - f_{\mathbf{w}}(\mathbf{x}^{(i)})\right)^2$$



2.1 Introduction. Generalization and overfitting

This is an example of 3-Fold Cross Validation

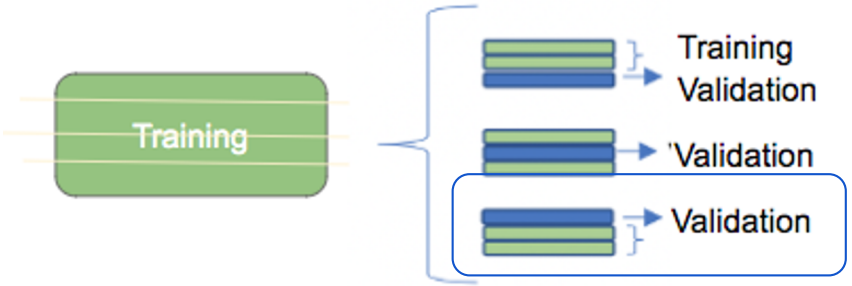




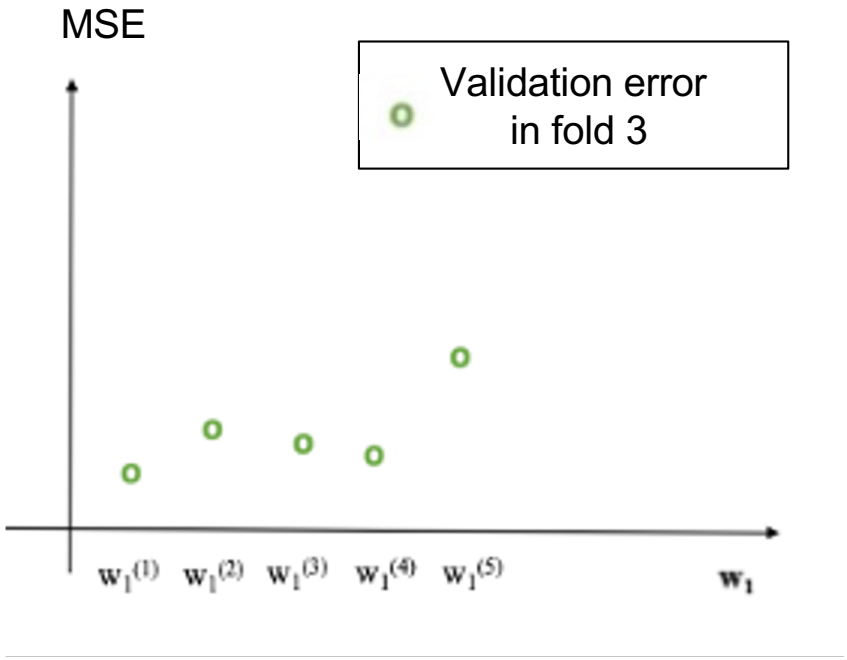
$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - f_{\mathbf{w}}(\mathbf{x}^{(i)})\right)^2$$

2.1 Introduction. Generalization and overfitting

This is an example of 3-Fold Cross Validation

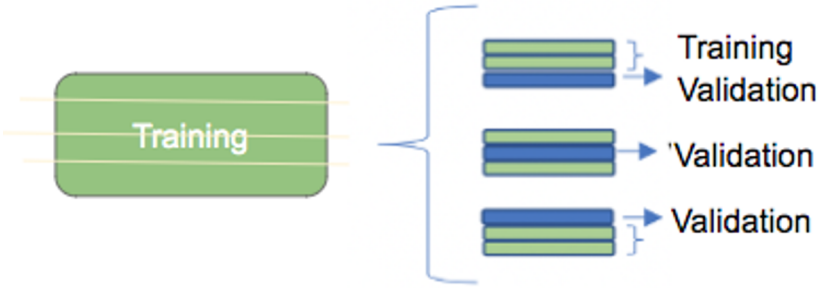


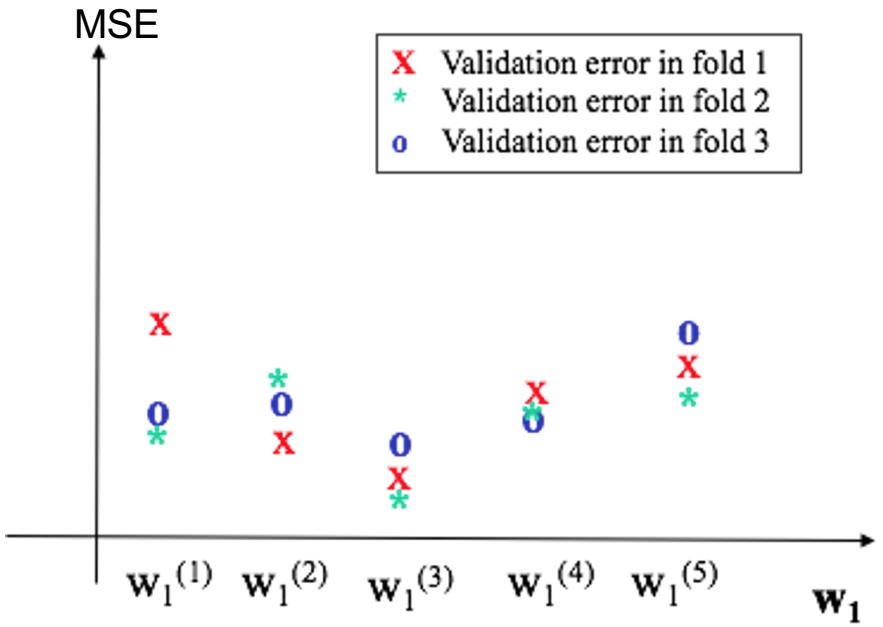
$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - f_{\mathbf{w}}(\mathbf{x}^{(i)})\right)^2$$



2.1 Introduction. Generalization and overfitting

This is an example of 3-Fold Cross Validation

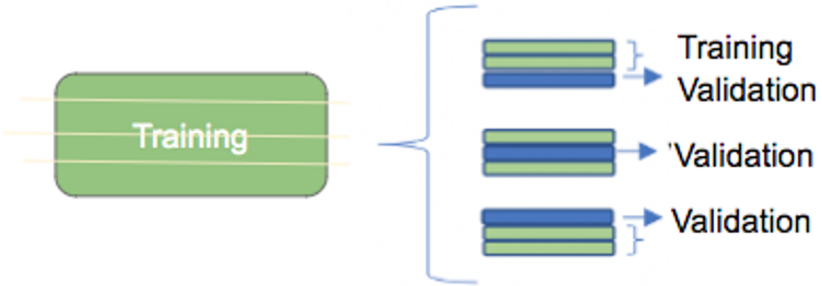




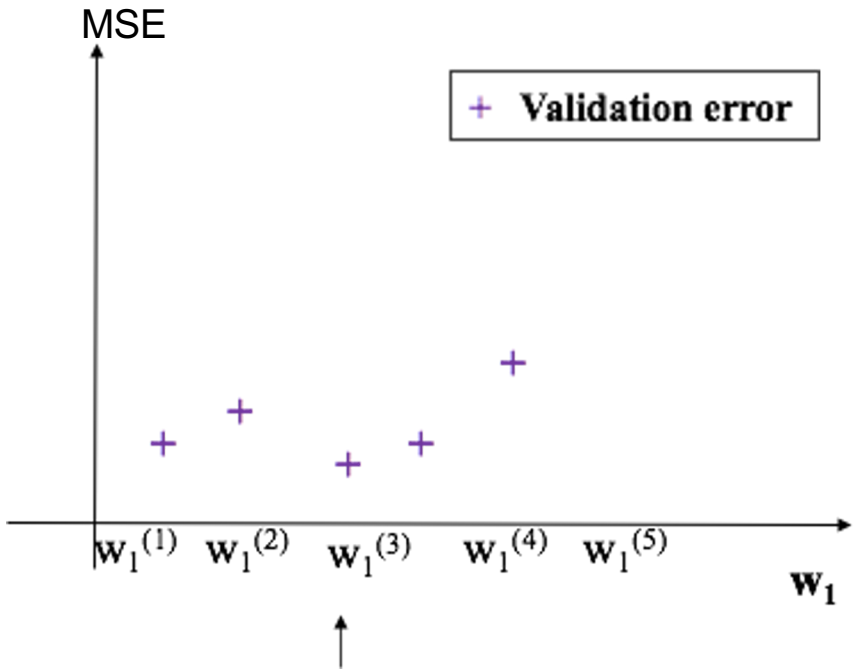
$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - f_{\mathbf{w}}(\mathbf{x}^{(i)})\right)^2$$

2.1 Introduction. Generalization and overfitting

This is an example of 3-Fold Cross Validation



$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - f_{\mathbf{w}}(\mathbf{x}^{(i)})\right)^2$$



2.1 Introduction. Generalization and overfitting

This is an example of 3-Fold Cross Validation

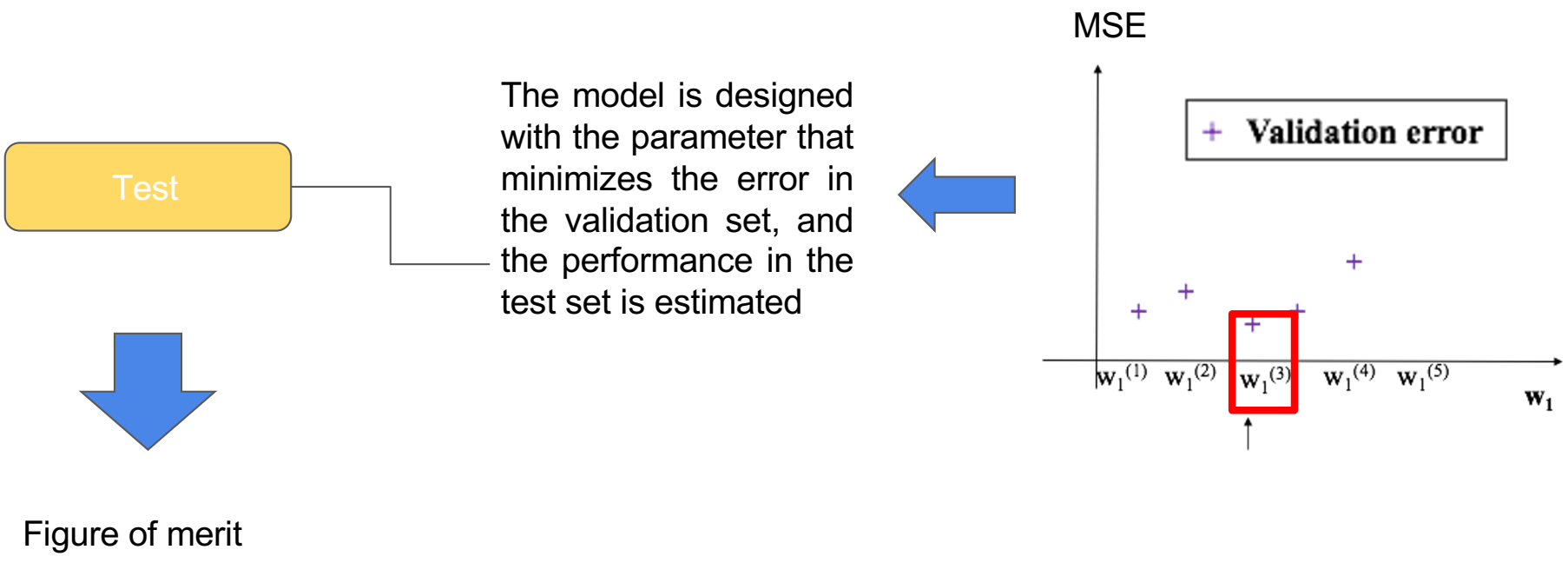
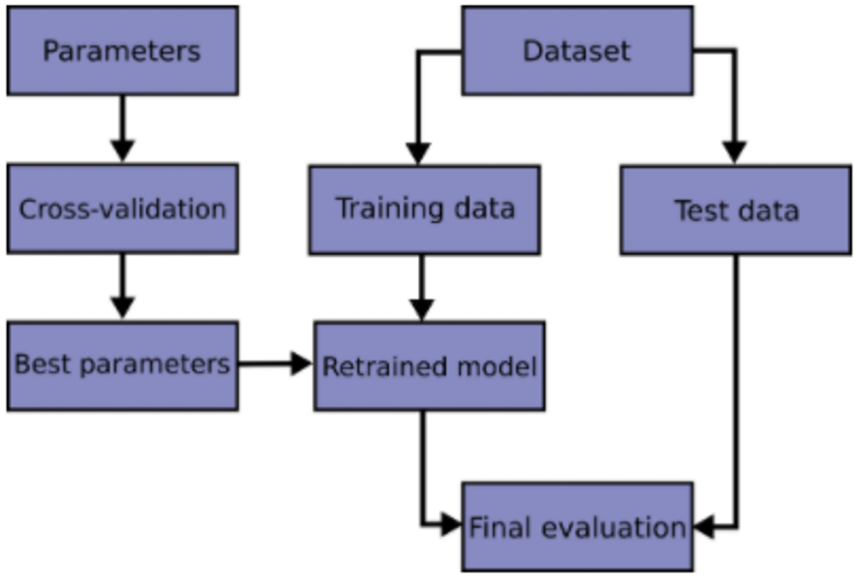


Figure of merit

2.1 Introduction. Generalization and overfitting

```
model = PolynomialRegression()

degrees = np.arange(1, 25)

cv_model = GridSearchCV(estimator,

                        param_grid={'deg': degrees},

                        scoring='mean_squared_error')

 cv_model.fit(xs, ys);

 cv_model.best_params_, cv_model.best_estimator_.coef_

 cv_model.predict
```



2.1 Introduction. Metric in regression

Mean Absolut Error

$$MAE = \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})$$

Mean Square Error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2$$

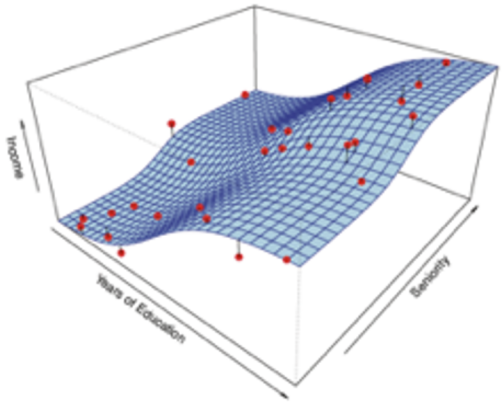
The Root of Mean Square Error

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2}$$

Contents

2.2 Univariate and Multivariate Regression

Data and underlying model (data generator)



Should there be zero error?

Why estimate the model? To perform prediction and inference tasks. *Estimate the relationship between variables*

In prediction tasks,
Input ->  -> Output

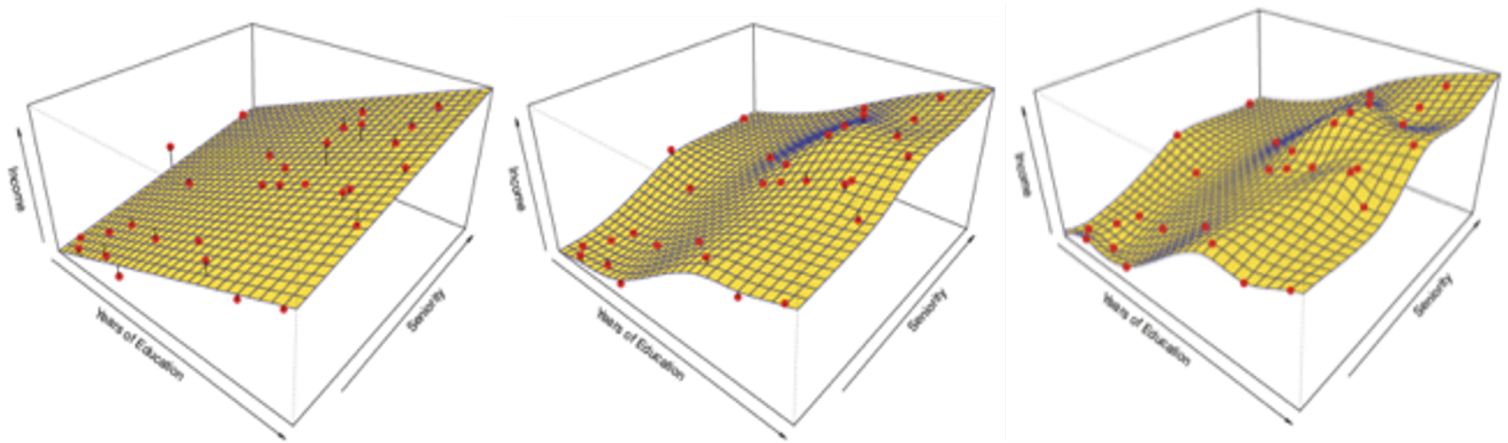
We have the entrance to the model (e.g. years of study and seniority), and we want to estimate the exit (e.g. income).

In **inference tasks**:

1.- Identify the variables that **most influence** the prediction (type of room? age? gender?)
2.- Know the relationship (positive or negative) between the output and each predictor variable

2.2 Univariate and Multivariate Regression

Parametric versus non-parametric models



Parametric linear model | Non-parametric models | Overfitting

$$\texttt{income} \approx \beta_0 + \beta_1 \times \texttt{education} + \beta_2 \times \texttt{seniority}.$$

Can there be non-linear parametric models?

2.2 Univariate and Multivariate Regression

Parametric versus non-parametric models

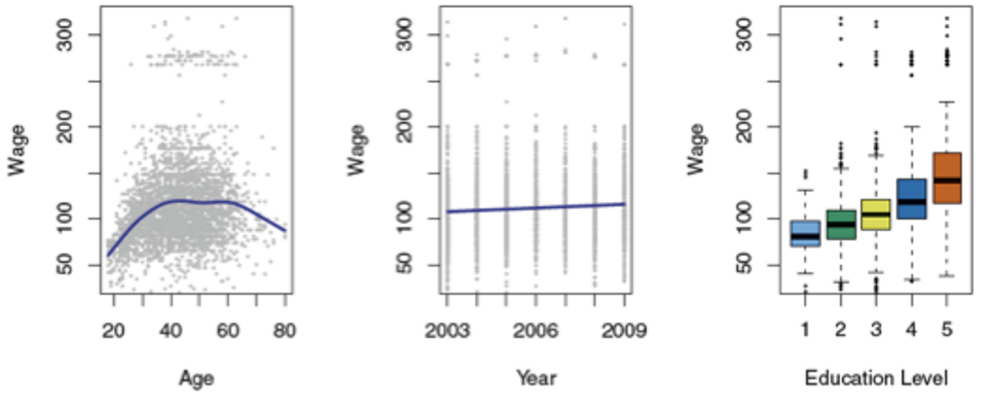
Non-parametric models do not make any assumptions about the functional form of the model (linear, quadratic, ...). Instead, they look for a model that approximates the data as closely as possible (without under- or over-adjustment).

Non-parametric models, ... don't they have parameters? → Cross-validation

Understanding the association between predictor variable (input to the model) and model response (output) can be more complicated in overly flexible models. Overfitting

2.2 Univariate and Multivariate Regression

Objective: To estimate the wage (continuous variable) of workers in a given region of the United States (3000 observations) regression task. Several factors are considered that may be related to salary (e.g. age of the worker, current year, educational level).



"An introduction to statitistical learning with applications in R". Autores: G. James, D. Witten, T. Hastie, R. Tibshirani. Editorial: Springer 2013.

A lot of variability is observed by considering each variable independently.

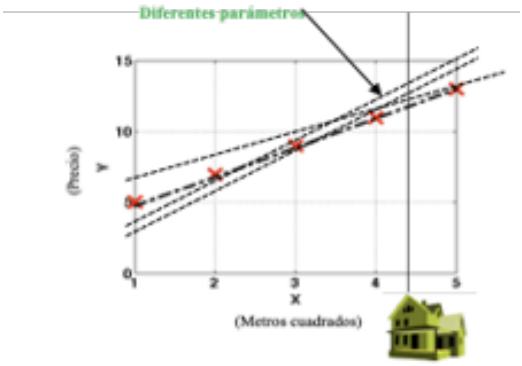
Multivariate analysis

2.2 Univariate and Multivariate Regression

In many disciplines, it is very important to know the relationship or dependence between variables. **For example**, is there a relationship between an individual's level of education and the salary received, and between weight and body mass index? Univariate regression

Sometimes, it is also important to know the effect that some variables can cause on another variable. **For example**, does the salary received by a worker depend on an individual's level of education and age? Is this relationship linear or non-linear? Multivariate regression

Sometimes the value of one variable can also be predicted in relation to others. **For example**, can you know the prices of a house if you know the square meters it has?



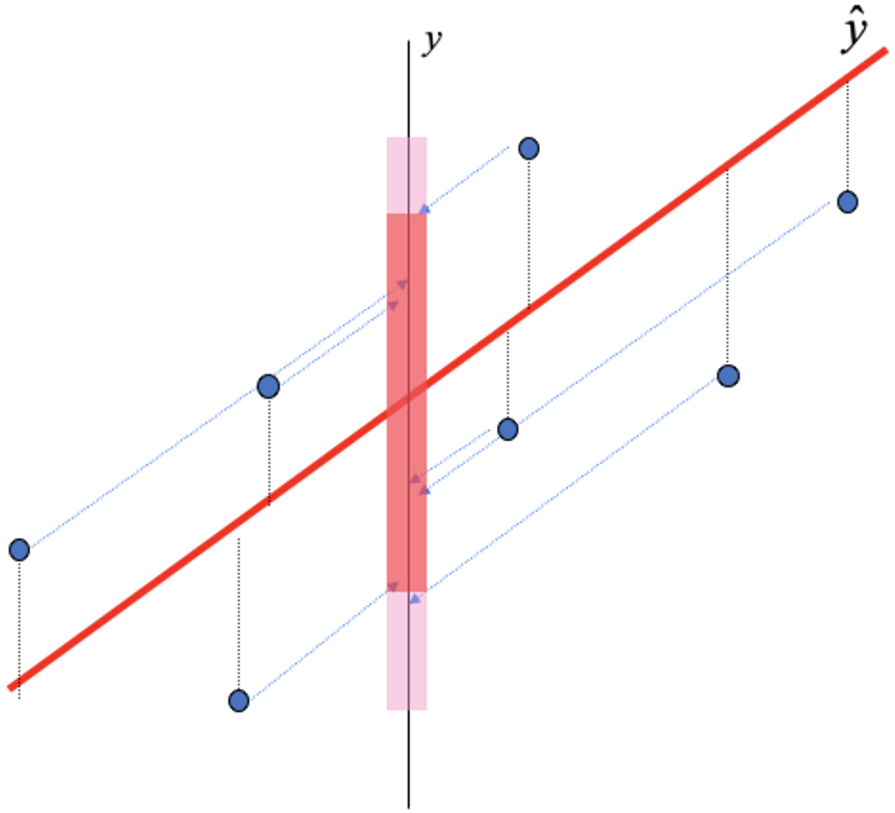
2.2 Univariate and Multivariate Regression

Linear adjustment (simple regression)

Estimation errors: vertical lines (difference between real value and value estimated by the model)

When we project the residues on the "y-axis", we observe that they are less dispersed than the original "y" variable.

The less dispersed the residues are, the better the goodness of the model.



2.2 Univariate Regression

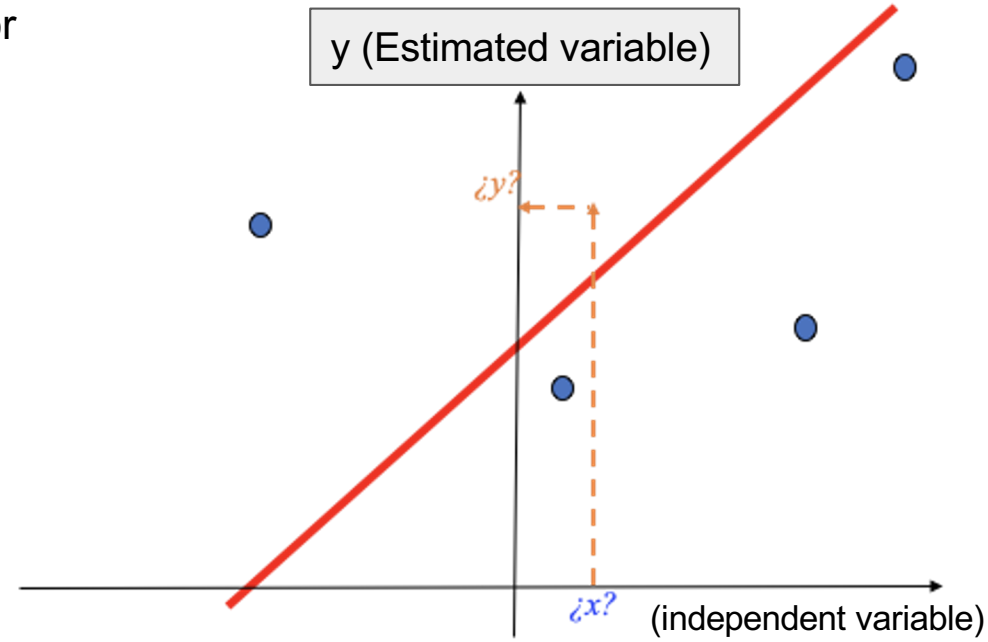
Objective: To specify the linear relationship between a set of N variables from K observations

$$\left\{\underline{\mathbf{x}}^{(k)}, y^{(k)}\right\}_{k=1}^{K}$$

Goal: To estimate the value of the variable **y** for new observations of **x**

$$\hat{y} = w_0 + w_1 x$$

The problem is reduced to finding the values of the weight parameters (w_0 and w_1 in the case of a single independent variable)

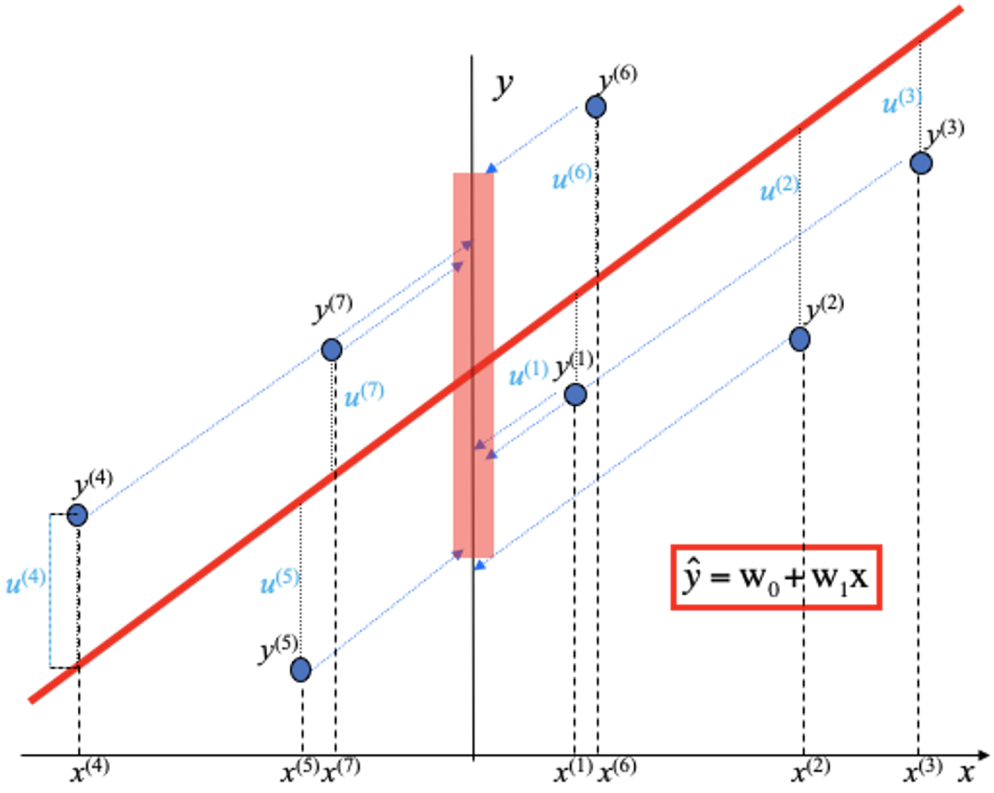


2.2 Univariate Regression

In regression, the error made in the estimations is called residual. The **residual** corresponds to the difference between the real value and the value estimated by the model - which is linear in the case of linear regression.

$$\left\{u^{(k)}\right\}_{k=1}^{K}$$

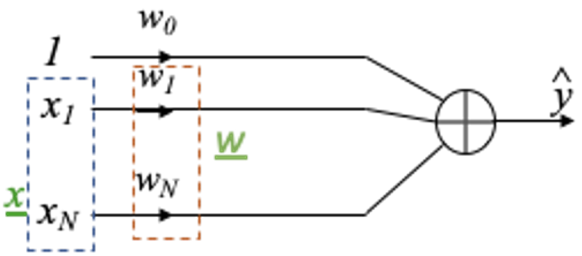
In regression analysis, the residual is denoted by the letter u (uncertainty) there are as many residuals as there are observations



2.2 Multivariate Regression

Estimation of the random variable and from the linear combination of N random variables x_i, i=1, ..., N find the parameters of the linear combination.

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_N x_N = w_0 + \underline{w}^T \underline{x} = \underline{w}_e^T \underline{x}_e$$



$$\left(\underline{x}_e = \begin{bmatrix} 1 \\ \underline{x} \end{bmatrix}; \underline{w}_e = \begin{bmatrix} w_0 \\ \underline{w} \end{bmatrix} \right)$$

x: feature vector
$\mathbf{x}_e$: extended feature vector

To find the w_i parameters, we minimize the mean square error

$$\hat{\underline{w}}_e : \min_{\underline{w}_e} E\left[\left(y - \hat{y} \right)^2 \right] = \min_{\underline{w}_e} E\left[e^2 \right]$$

2.2 Multivariate Regression

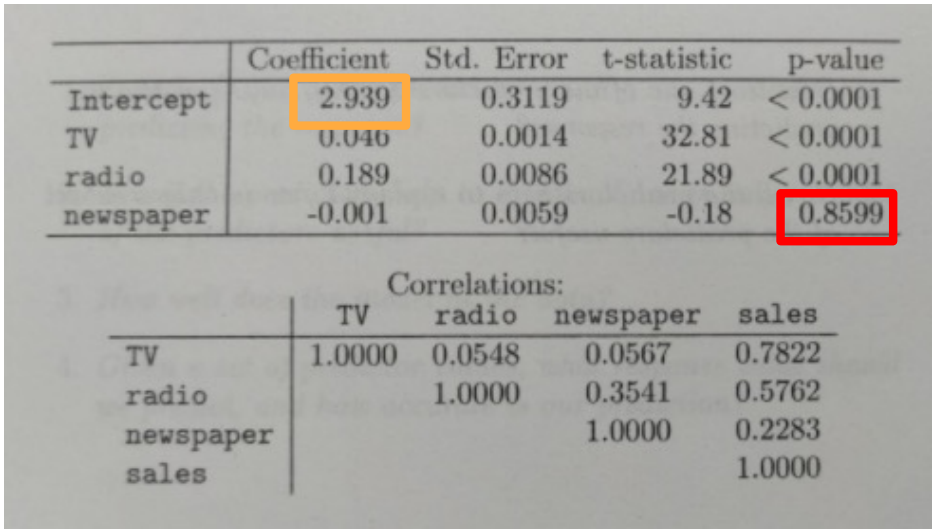
Interpreting regression coefficients

- Coefficients should be uncorrelated. If this occurs:
  - * Each coefficient can be estimated and tested separately
  - * The following interpretation is possible: “A unit change in $\mathbf{x}_j$ is related to a w_j change in **y,** while the other variables stay fixed

- When features are correlated:
  - * The variance of all coefficients can increase.
  - * Interpreting regression coefficients become more difficult

- Important! Correlation does not imply causality!

2.2 Multivariate Regression



	Coefficient	Std. Error	t-statistic	p-value
Intercept	2.939	0.3119	9.42	< 0.0001
TV	0.046	0.0014	32.81	< 0.0001
radio	0.189	0.0086	21.89	< 0.0001
newspaper	-0.001	0.0059	-0.18	0.8599

Correlations:

	TV	radio	newspaper	sales
TV	1.0000	0.0548	0.0567	0.7822
radio		1.0000	0.3541	0.5762
newspaper			1.0000	0.2283
sales				1.0000

Example obtained from “An introduction to statitistical learning with applications in R”. Autores: G. James, D. Witten, T. Hastie, R. Tibshirani. Editorial: Springer 2013.

2.2 Univariate and Multivariate Regression

Below we will describe two coefficients that allow us: (1) to quantify the degree of linear association between the variables analyzed, and (2) to measure the goodness of the adjustment of the linear regression obtained from a set of observed data.

The linear correlation coefficient

It measures the linear relationship between the dependent variable (x) and the independent variable (y). It is a non-dimensional measure:

$$r = \frac{\sum_{k=1}^{K}(x^{(k)} - \bar{\mathbf{x}})(y^{(k)} - \bar{\mathbf{y}})}{\sqrt{\sum_{k=1}^{K}(x^{(k)} - \bar{\mathbf{x}})^2}\sqrt{\sum_{k=1}^{K}(y^{(k)} - \bar{\mathbf{y}})^2}}$$

Take values between -1 and 1

2.2 Univariate and Multivariate Regression

The linear correlation coefficient

It measures the linear relationship between the dependent variable (x) and the independent variable (y). It is a non-dimensional measure:

$$r = \frac{\sum_{k=1}^{K}(x^{(k)} - \bar{\mathbf{x}})(y^{(k)} - \bar{\mathbf{y}})}{\sqrt{\sum_{k=1}^{K}(x^{(k)} - \bar{\mathbf{x}})^2}\sqrt{\sum_{k=1}^{K}(y^{(k)} - \bar{\mathbf{y}})^2}}$$

Take values between -1 and 1

If **r=0**, it indicates that the linear relationship between variables is little or none.
If **r=1**, there is an exact linear relationship between variables.
If **r=-1**, there is an exact linear relationship between variables but they vary in the opposite direction.

Values closer in absolute value to 1, indicate that the greater the degree of linear association between variables

2.2 Univariate and Multivariate Regression

Determination coefficient

It measures the goodness of the adjustment made with the *linear regression*. Quantifies the proportion of variation of the variable Y that is explained by the variable X (dependent variable).

$$R^2 = \frac{(\hat{y}^{(k)} - \bar{\mathbf{y}})^2}{(y^{(k)} - \bar{\mathbf{y}})^2}$$

$$R^2 = r^2$$

The higher the value of R^2, the better the fit, and therefore the greater the reliability of the predictions made with that model.

If $R^2 = r^2 = 1$, the dependent variable explains all the variation of the variable Y, and therefore there would be no error in the predictions made.

2.2 Univariate and Multivariate Regression

```
import numpy as np
    from sklearn.linear_model import LinearRegression
model = LinearRegression()
# Fit the model with training data
model = LinearRegression().fit(x, y)

# Print the model intercept and coefficients
print('intercept:', model.intercept_)
print('coefficients:', model.coef_)

# Predict - use test data
y_pred = model.predict(x)

# Print the predicted output
print('predicted output:', y_pred, sep='\n')

# Calculate R^2
r_2=model.score(x, y)

RMSE_linear = np.sqrt(np.sum(np.square(y_pred-y_linear)))
```

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

Examples

```
>>> import numpy as np
>>> from sklearn.linear_model import LinearRegression
>>> X = np.array([[1, 1], [1, 2], [2, 2], [2, 3]])
>>> # y = 1 * x_0 + 2 * x_1 + 3
>>> y = np.dot(X, np.array([1, 2])) + 3
>>> reg = LinearRegression().fit(X, y)
>>> reg.score(X, y)
1.0
>>> reg.coef_
array([1., 2.])
>>> reg.intercept_
3.0000...
>>> reg.predict(np.array([[3, 5]]))
array([16.])
```

Contents

2.3 Non-linear regression

Many of the relationships between variables we study are **non-linear**. We can highlight the functions logarithmic, inverse, quadratic, cubic, power, exponential, etc.

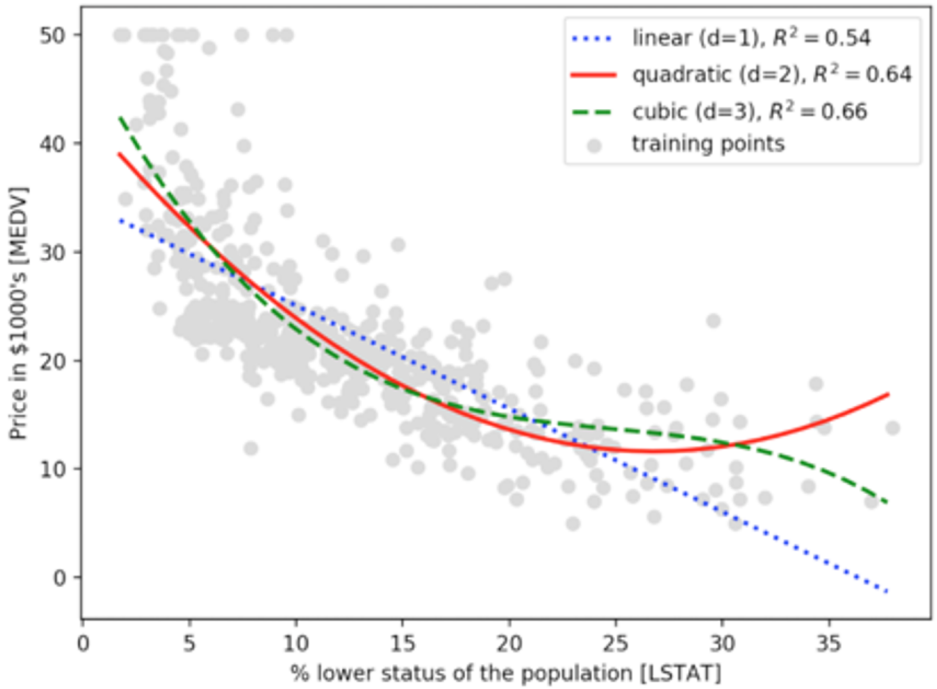
Some examples:

a) Exponential model: $\hat{y} = w_o + e^{bx}$

a) Polynomial model: $\hat{y} = w_0 + w_1 x + ... + w_m x^m$

$$\hat{y} = w_0 + w_1 x_1^2 + w_2 x_1^3 + w_3 x_2^2$$

2.3 Non-linear regression



2.3 Non-linear regression

Polynomial models can be adjusted by linear least-squares regression because, although they generate non-linear models, their equation is still a linear equation with predictors.

How to select the degree of the polynomial?

Cross-validation

The use of polynomial models with a grade greater than 3 or 4 is not recommended due to excessive flexibility → overfitting

2.3 Non-linear regression

```
# Fitting Polynomial Regression to the dataset

from sklearn.preprocessing import PolynomialFeatures
poly_reg = PolynomialFeatures(degree=4)   # We can find the degree by cross validation
X_poly = poly_reg.fit_transform(X)
pol_reg = LinearRegression()
# Fit the model with training data
pol_reg.fit(X_poly, y)

# Plot the results
plt.scatter(X, y, color='red')
# We predict the results for the test set
plt.plot(X, pol_reg.predict(poly_reg.fit_transform(X)), color='blue')
plt.show()
```

2.3 Non-linear regression

- poly = PolynomialFeatures(2)

  The features of X have been transformed from (X_1, X_2) to $(1, X_1, X_2, X_1^2, X_1X_2, X_2^2)$.

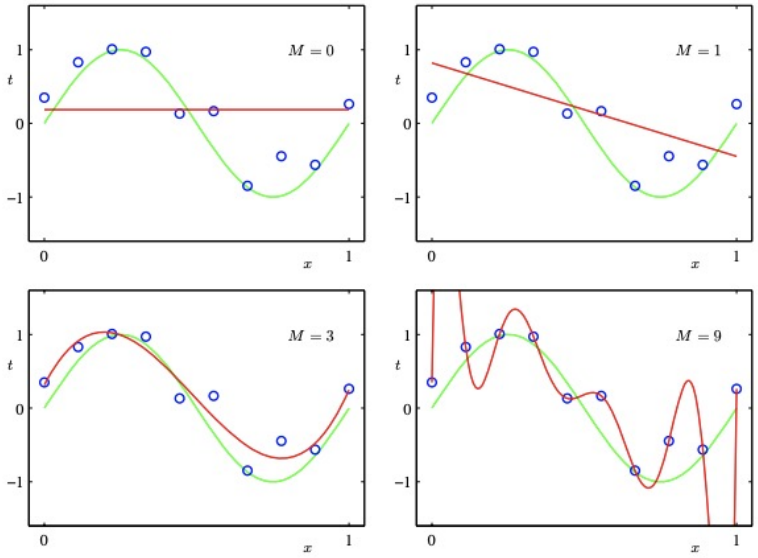
- poly = PolynomialFeatures(3, interaction_only=**True**)

The features of X have been transformed from (X_1, X_2, X_3) to $(1, X_1, X_2, X_3, X_1X_2, X_1X_3, X_2X_3, X_1X_2X_3)$.

Contents

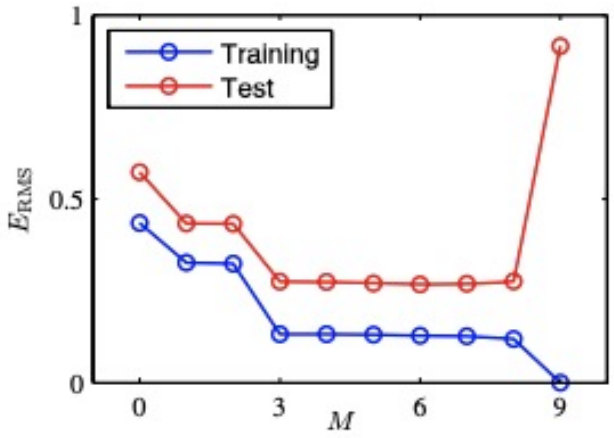
2.4 Linear regression with regularization

- As expected, the use of high level polynomials causes the equation to fit itself to the noise in the data.



- **Heuristic:** The number of samples has to > 10 times number of variables.

2.4 Linear regression with regularization

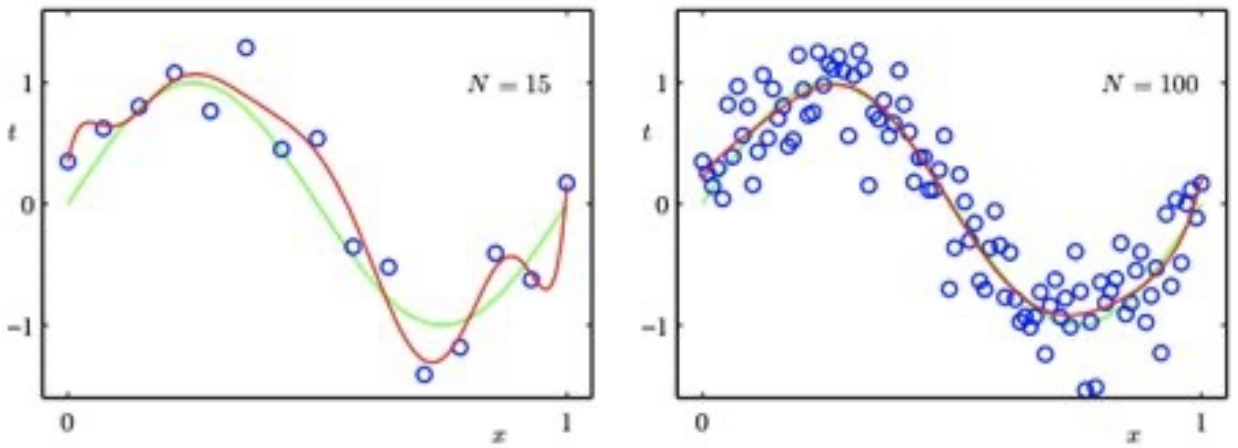


EMRS = RMSE (Root Mean Square Error)

Note that for M = 9, the error in the training set is zero. There is 10 degrees of freedom corresponding to the 10 coefficients $w_0,...,w_9$

Then, we can tunned exactly them to the 10 data points in the training set

2.4 Linear regression with regularization



It can be seen that for a given model, the overfitting problem become less severe as the size of the data set increases.

Heuristic: The number of samples has to > 10 times number of variables.

2.4 Linear regression with regularization

How to control overfitting?

Using regularization --> adding a penalty term to the error function in order to discourage the coefficients from reaching large values.

$$J(\mathbf{w}) = \frac{1}{2m}\sum_{i=1}^{m}(f(\mathbf{x})^{(i)} - y^{(i)})^2 + \frac{\lambda}{2}||\mathbf{w}||^q$$

The parameter λ is use as a trade-off between the sum of square error term and regularization term

$||\cdot||$ represents the l_q norm of the weigth vector

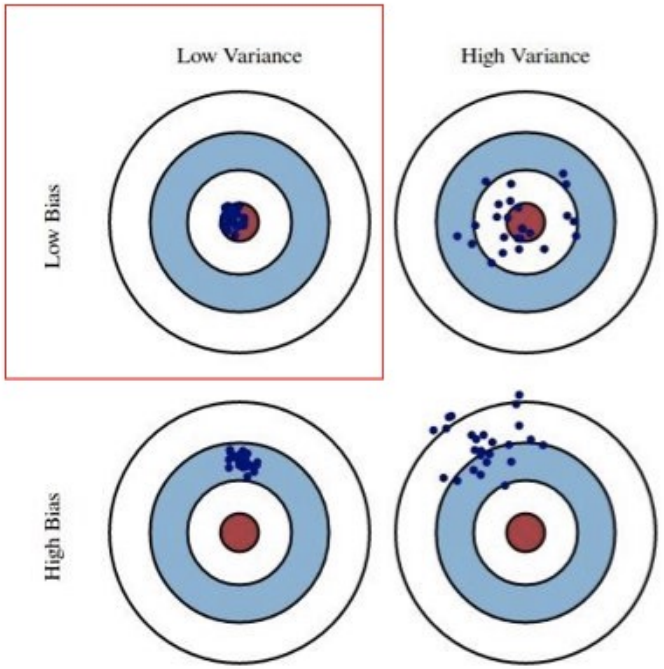
2.4 Linear regression with regularization

The parameter λ is use as a trade-off between

- The square error term
- The regularization term

When λ

• Increases more coefficients are set to zero and eliminated
• Theoretically, when λ = ∞, *all* coefficients are eliminated).
•As λ increases, bias increases.
•As λ decreases, variance decreases.

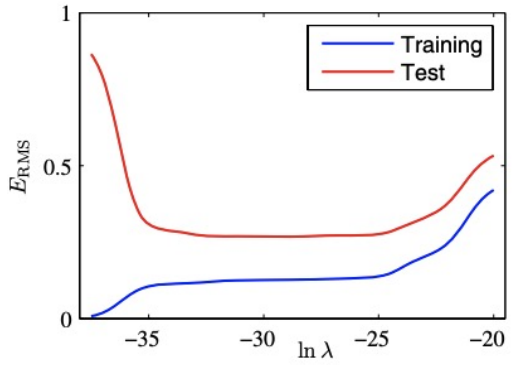


2.4 Linear regression with regularization

Ridge regression

The particular case of a quadratic regularizer is called ridge regression. This forces the model weigths to be as small as possible as well as fit the data.

$$J(\mathbf{w}) = \frac{1}{2m}\sum_{i=1}^{m}(f(\mathbf{x})^{(i)} - y^{(i)})^2 + \frac{\lambda}{2}||\mathbf{w}||^2$$



2.4 Linear regression with regularization

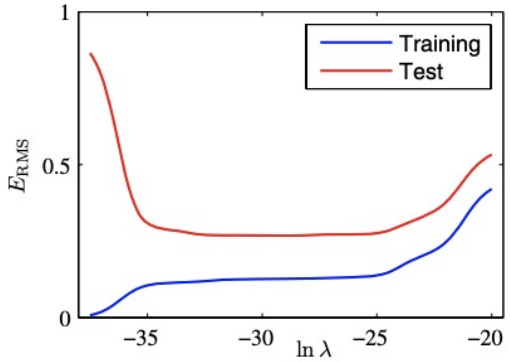
Lasso

The case of q = 1 is called Least absolute shrinkage and selection operator (LASSO).

Note that if λ is large enough, some of the coefficients w_j are driven to zero and, therefore, some of the basis function are not considered, achieving a sparse model. LASSO automatically performs feature selection

$$J(\mathbf{w}) = \frac{1}{2m}\sum_{i=1}^{m}(f(\mathbf{x})^{(i)} - y^{(i)})^2 + \frac{\lambda}{2}||\mathbf{w}||^1$$

$$||\mathbf{w}||^1 < B$$



https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html

2.4 Linear regression with regularization

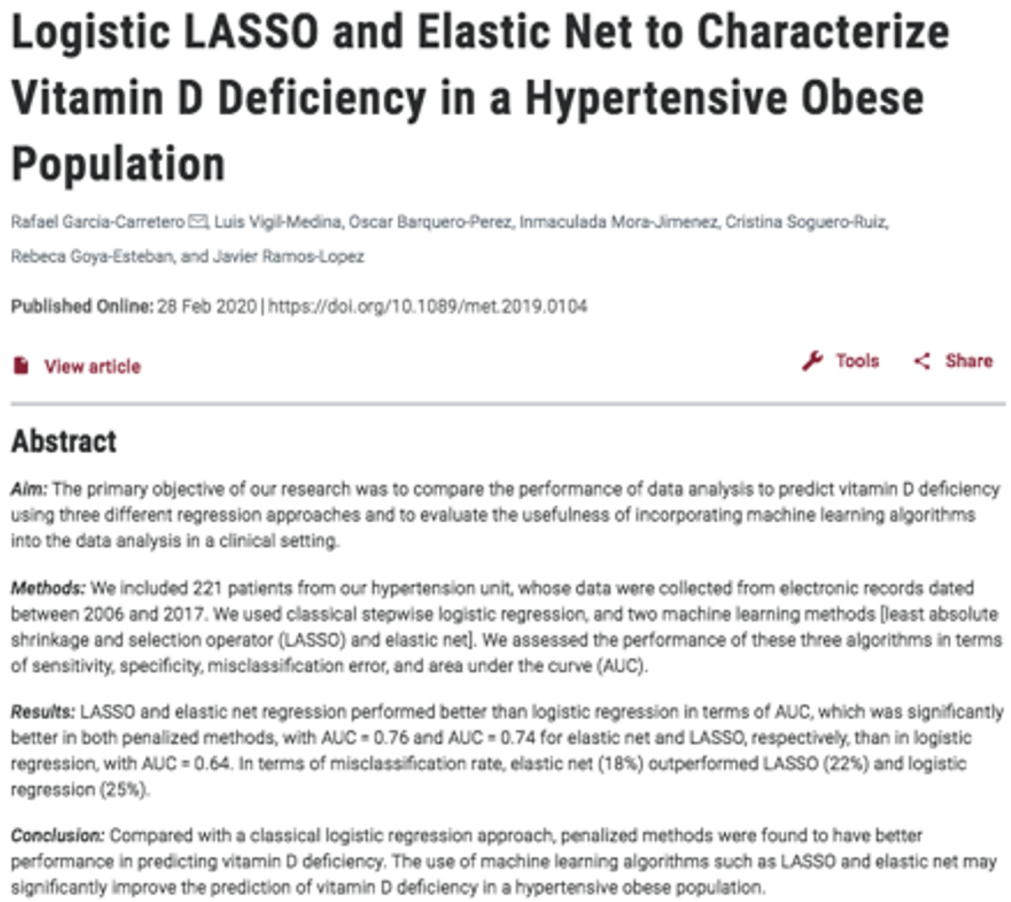
Elastic net

It is a combination of lasso and ridge regression. It provides a good combination of sparsity and regularization:

$$J(\mathbf{w}) = \frac{1}{2m}\sum_{i=1}^{m}(f(\mathbf{x})^{(i)} - y^{(i)})^2 + \lambda_2||\mathbf{w}||^2 + \lambda_1||\mathbf{w}||^1$$

Contents

2.6 Biomedical examples and application



Logistic LASSO and Elastic Net to Characterize Vitamin D Deficiency in a Hypertensive Obese Population

Rafael Garcia-Carretero, Luis Vigil-Medina, Oscar Barquero-Perez, Inmaculada Mora-Jimenez, Cristina Soguero-Ruiz, Rebeca Goya-Esteban, and Javier Ramos-Lopez

Published Online: 28 Feb 2020 | https://doi.org/10.1089/met.2019.0104

View article Tools Share

Abstract

Aim: The primary objective of our research was to compare the performance of data analysis to predict vitamin D deficiency using three different regression approaches and to evaluate the usefulness of incorporating machine learning algorithms into the data analysis in a clinical setting.

Methods: We included 221 patients from our hypertension unit, whose data were collected from electronic records dated between 2006 and 2017. We used classical stepwise logistic regression, and two machine learning methods [least absolute shrinkage and selection operator (LASSO) and elastic net]. We assessed the performance of these three algorithms in terms of sensitivity, specificity, misclassification error, and area under the curve (AUC).

Results: LASSO and elastic net regression performed better than logistic regression in terms of AUC, which was significantly better in both penalized methods, with AUC = 0.76 and AUC = 0.74 for elastic net and LASSO, respectively, than in logistic regression, with AUC = 0.64. In terms of misclassification rate, elastic net (18%) outperformed LASSO (22%) and logistic regression (25%).

Conclusion: Compared with a classical logistic regression approach, penalized methods were found to have better performance in predicting vitamin D deficiency. The use of machine learning algorithms such as LASSO and elastic net may significantly improve the prediction of vitamin D deficiency in a hypertensive obese population.

2.6 Biomedical examples and application

The L1 penalty is the sum of the absolute coefficients (w_j):

$$|w|_1 = \sum_{j=1}^{p} |w_j|.$$

LASSO uses this L1 penalty by adding λ to control the penalization:

$$\hat{w} = arg \min_{w} \sum_{i=1}^{n} \left(y_i - \sum_j x_{ij} w_j \right)^2 + \lambda \sum_{j=1}^{p} |w_j|.$$

The L2 norm is the sum of the square of the coefficients (w_j),

$$|x| = \sum_{j=1}^{p} w_j^2,$$

which is used by ridge regression:

$$\hat{w} = arg \min_{w} \sum_{i=1}^{n} \left(y_i - \sum_j x_{ij} w_j \right)^2 + \lambda \sum_{j=1}^{p} w_j^2.$$

Elastic net uses both penalties, combines the advantages of them and may solve their limitations. Therefore, it has the effect of shrinking coefficients (as in ridge regression) and setting some coefficients to zero (as in LASSO), and thus automatically selecting features. The estimates from the elastic net algorithm are defined by

$$\hat{w} = arg \min_{w} \sum_{i=1}^{n} \left(y_i - \sum_j x_{ij} w_j \right)^2 + \lambda_1 \sum_{j=1}^{p} |w_j| + \lambda_2 \sum_{j=1}^{p} w_j^2.$$

TABLE 2. FEATURE SELECTION USING DIFFERENT APPROACHES WITHIN OUR COHORT

	Stepwise logistic regression	*Logistic LASSO*	*Elastic net*
Gender (women)			
Age (years)	–0.045	–0.012	–0.008
Waist circumference (cm)			
BMI			
Metabolic syndrome			
SBP (mmHg)	0.022	0.005	0.003
DBP (mmHg)		0.004	0.005
Uric acid (mg/dL)			
Total cholesterol (mg/dL)	0.009		
Triglycerides (mg/dL)			
LDL cholesterol (mg/dL)		0.003	0.002
HDL cholesterol (mg/dL)			
Blood glucose (mg/dL)			
HbA1c (%)	0.94		
HOMA-IR			
Ferritin (mg/dL)			
GGT (mg/dL)	–0.014	–0.0002	
CRP	0.114	0.069	0.055
Creatine (mg/dL)			
Cystatin C (mg/dL)			

LASSO, least absolute shrinkage and selection operator.

2.6 Biomedical examples and application

Table 2. Feature Selection Using Different Approaches Within Our Cohort

	Stepwise logistic regression	*Logistic LASSO*	*Elastic net*
Gender (women)			
Age (years)	−0.045	−0.012	−0.008
Waist circumference (cm)			
BMI			
Metabolic syndrome			
SBP (mmHg)	0.022	0.005	0.003
DBP (mmHg)		0.004	0.005
Uric acid (mg/dL)			
Total cholesterol (mg/dL)	0.009		
Triglycerides (mg/dL)			
LDL cholesterol (mg/dL)		0.003	0.002
HDL cholesterol (mg/dL)			
Blood glucose (mg/dL)			
HbA1c (%)	0.94		
HOMA-IR			
Ferritin (mg/dL)			
GGT (mg/dL)	−0.014	−0.0002	
CRP	0.114	0.069	0.055
Creatine (mg/dL)			
Cystatin C (mg/dL)			

LASSO, least absolute shrinkage and selection operator.

Table 3. Performance of the Different Algorithms Using the Testing Data Regarding Vitamin D Deficiency

	Sensitivity	*Specificity*	*AUC*	*Misclassification error*
Logistic regression	66%	44%	0.64	0.25
LASSO	71%	40%	0.74	0.22
Elastic net	76%	50%	0.76	0.18

Bibliography

•Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

•James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112, p. 18). New York: Springer.

• Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.