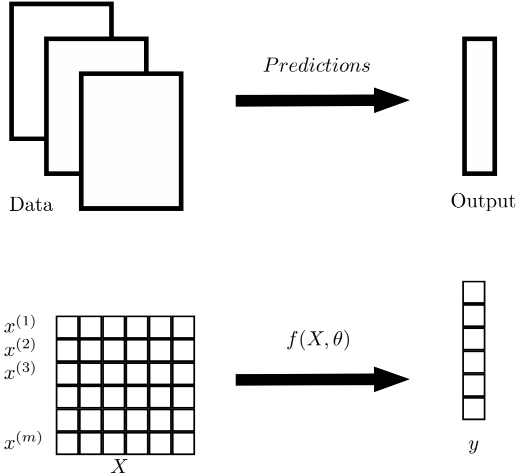
BABD

Masters in Business Analytics and Big Data

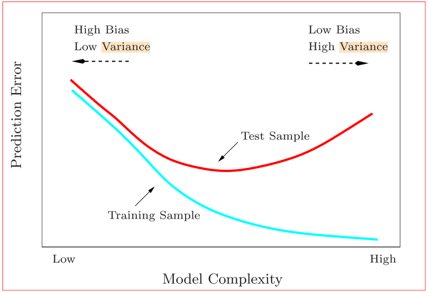
Regression

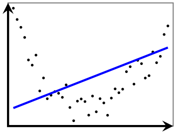
Mauricio Soto - mauricioabel.soto@polimi.it

Supervised Learning

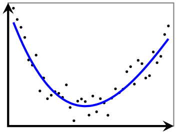


Under/Over-fitting

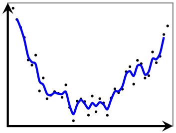




Underfitting



Balance



Overfitting

Quality measures - Regression

- Coefficient of determination

$$R^2 = 1 - \sum_{i=1}^{m}(y_i - \hat{y}_i)^2 / \sum_{i=1}^{m}(y_i - \bar{y})^2$$

- Mean Absolute Error :

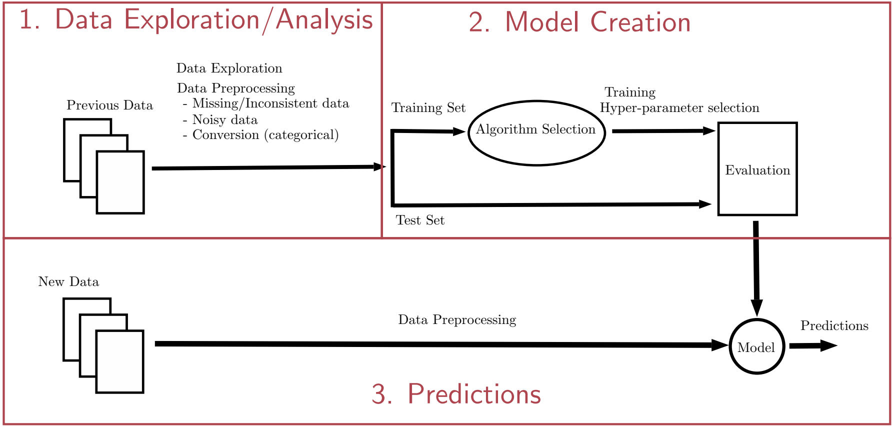
$$MAE = \frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i|$$

- Mean Squared Error :

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

- Root Mean Squared Error : $RMSE = \sqrt{MSE}$

Supervised Learning Workflow



Regression Models

- Heuristics Methods
  - Nearest Neighbours
  - Regression Trees
- Optimization based Methods
  - Linear models
  - Support vector machine
  - Neural Networks

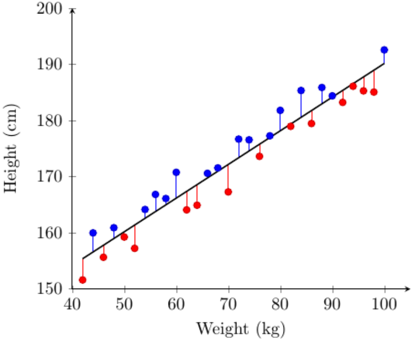
Simple linear regression

- Deterministic model

$$Y = w\,X + b$$

- Probabilistic model

$$Y = w\,X + b + \varepsilon$$



Regression models (n=1)

- linear

$$Y = b + \sum_{j=1}^{n} w_j X_j \; = \; b + w_1 X_1 + w_2 X_2 + \cdots + w_n X_n \; = \; b + Xw$$

- quadratic

$$\begin{aligned} Y &= b + Xw + X^2 d \qquad && Z = X^2 \\ &= b + Xw + Zd \end{aligned}$$

- exponential

$$\begin{aligned} Y &= e^{b+Xw} \qquad && Z = logY \\ &= b + Xw \end{aligned}$$

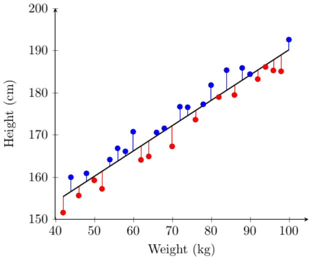
Simple linear regression

- Residuals

$$e_i = y_i - f(x_i) = y_i - wx_i - b \qquad i \in \mathcal{M}$$

- Least square regression

$$SSE = \sum_{i=1}^{m} e_i^2 = \sum_{i=1}^{m} [y_i - wx_i - b]^2$$



Least square linear regression

$$\frac{\partial SSE}{\partial b} = -2\sum_{i=1}^{m}[y_i - wx_i - b] = 0 \Rightarrow \qquad w\sum_{i=1}^{m} x_i + bm = \sum_{i=1}^{m} y_i$$

$$\frac{\partial SSE}{\partial w} = -2\sum_{i=1}^{m}[y_i - wx_i - b]x_i = 0 \Rightarrow \qquad w\sum_{i=1}^{m} x_i^2 + b\sum_{i=1}^{m} x_i = \sum_{i=1}^{m} x_i y_i$$

Least square linear regression

$$\frac{\partial SSE}{\partial b} = -2\sum_{i=1}^{m}[y_i - wx_i - b] = 0 \Rightarrow \qquad w\sum_{i=1}^{m} x_i + bm = \sum_{i=1}^{m} y_i$$

$$\frac{\partial SSE}{\partial w} = -2\sum_{i=1}^{m}[y_i - wx_i - b]x_i = 0 \Rightarrow \qquad w\sum_{i=1}^{m} x_i^2 + b\sum_{i=1}^{m} x_i = \sum_{i=1}^{m} x_i y_i$$

$$w^* = \frac{\sigma_{xy}}{\sigma_{xx}}, \qquad b^* = \overline{\mu}_y - w^*\overline{\mu}_x$$

$$\sigma_{xx} = \sum_{i=1}^{m}(x_i - \overline{\mu}_x)^2$$

$$\sigma_{xy} = \sum_{i=1}^{m}(x_i - \overline{\mu}_x)(y_i - \overline{\mu}_y)$$

Least square multiple linear regression

- If we extend the matrix X with a vector of "ones" then the linear model can be expressed as

$$y = Xw + e$$

- $$SSE = \sum_{i=1}^{m} e_i^2 = ||e||^2 = (y - Xw)^\top (y - Xw)$$

- $$\nabla SSE = -2X^\top y + 2X^\top Xw = 0$$

- $$X^\top Xw = X^\top y$$

$$\boxed{w^* = (X^\top X)^{-1} X^\top y}$$

-

Least square multiple linear regression

- Solution:

$$w^* = (X^\top X)^{-1} X^\top y$$

- Predicted values

$$\hat{y} = Xw^* = (X(X^\top X)^{-1} X^\top) y = Hy$$

- Hat matrix

$$H = X(X^\top X)^{-1} X^\top, \qquad H^2 = H$$

- Residuals

$$e = y - \hat{y} = (I - H) y$$

General Linear Models

- We consider a set of bases functions: polynomials, kernels, etc.

$$Y = \sum_h w_h \, g_h(X_1, X_2, \dots, X_n) + b + \varepsilon$$

- For example, for $n = 2$

$$Y = X_1 w_1 + X_2 w_2 + X_1^2 w_3 + X_2^2 w_4 + [X_1 X_2] w_5 + b + \varepsilon$$

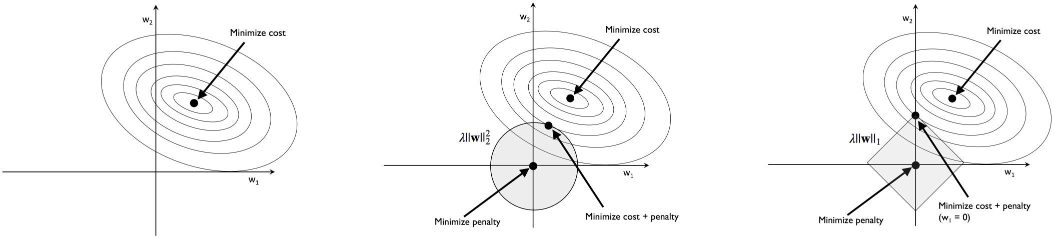
Linear Models Regularization

- Ridge:

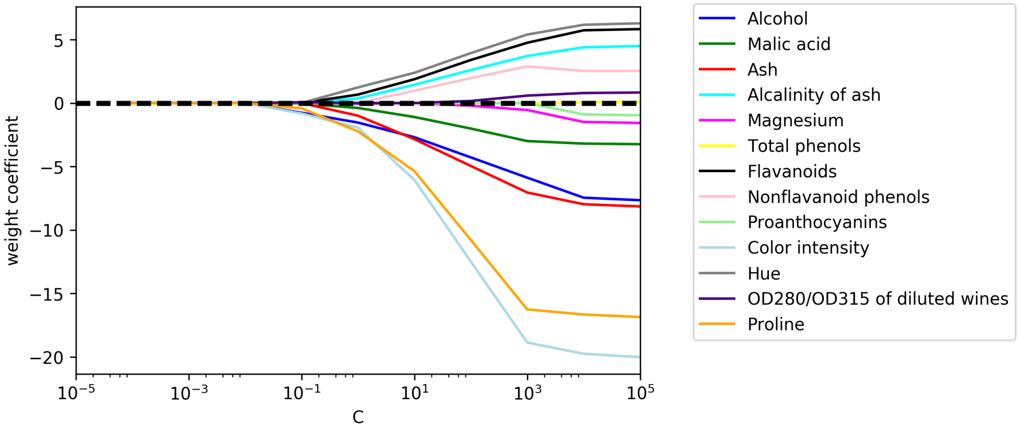
$$\min_{w} \lambda||w||^2 + ||e||^2 = \min_{w} \lambda||w||^2 + (y - Xw)^\top (y - Xw)$$

- Lasso:

$$\min_{w} \lambda|w| + ||e||^2 = \min_{w} \lambda|w| + (y - Xw)^\top (y - Xw)$$



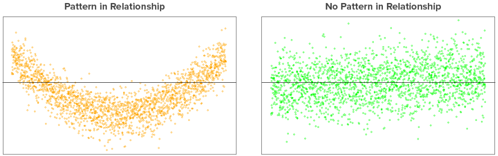
Regularization effect

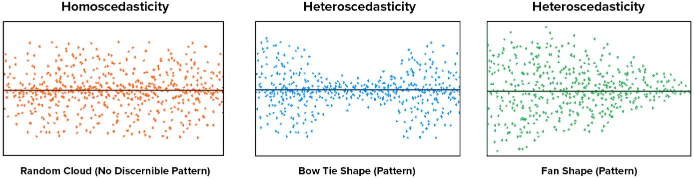


$$C = 1/\lambda$$

Residual assumptions

$$\text{Independence}, \quad E(\varepsilon_i|\mathbf{x_i}) = 0), \qquad Var(\varepsilon_i|\mathbf{x_i}) = \sigma^2$$





Linear models - Significance of coefficients

- By assuming residuals independent and normal distributio
- Variance of coefficients

$$Var(\hat{w}) = (X'X)^{-1}\sigma^2 \quad \hat{w} \sim \mathcal{N}(w, (X'X)^{-1}\sigma^2)$$

- Empirical Variance

$$\hat{\sigma} = \frac{SSE}{m-n-1} = \frac{\sum_{i=1}^{m}(y_i - \mathbf{w}'\mathbf{x_i})^2}{m-n-1} = \frac{\mathbf{y}'(\mathbf{I}-\mathbf{H})\mathbf{y}}{m-n-1}$$

-

$$(m-n-1)\,\hat{\sigma}^2 \sim \sigma^2\chi^2_{m-n-1}$$

- Under the null hypothesis $w_i = 0$ then

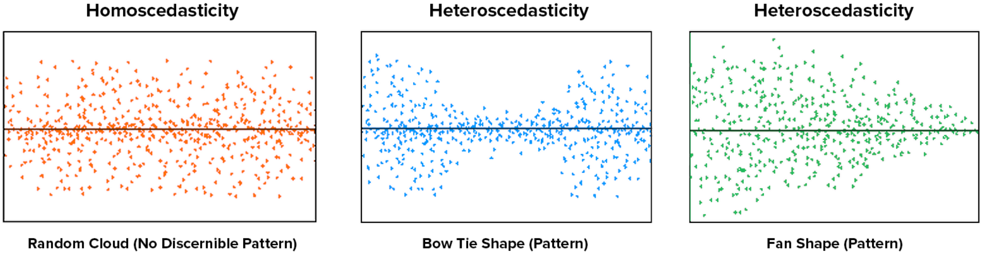
$$\frac{\hat{w}_i}{\hat{\sigma}\sqrt{(X'X)_{ii}}} \sim t_{m-n-1}$$

Linear models - Significance of coefficients

	coef	std err	t	P>|t|	[0.025	0.975]
const	22.5693	0.245	92.144	0.000	22.088	23.051
CRIM	-0.8678	0.298	-2.909	0.004	-1.455	-0.281
ZN	0.9310	0.365	2.551	0.011	0.213	1.649
INDUS	0.5166	0.494	1.045	0.297	-0.456	1.489
CHAS	0.0671	0.270	0.249	0.804	-0.463	0.598
NOX	-1.6601	0.532	-3.121	0.002	-2.706	-0.614
RM	3.3925	0.340	9.971	0.000	2.723	4.062
AGE	-0.2093	0.429	-0.488	0.626	-1.052	0.634
DIS	-2.7910	0.475	-5.879	0.000	-3.725	-1.857
RAD	2.3790	0.650	3.660	0.000	1.100	3.658
TAX	-2.1962	0.718	-3.059	0.002	-3.608	-0.784
PTRATIO	-2.0690	0.325	-6.372	0.000	-2.708	-1.430
B	0.5860	0.298	1.965	0.050	-0.001	1.173
LSTAT	-3.4712	0.432	-8.032	0.000	-4.321	-2.621

Normal residual assumption

- Graphical distribution



- Graphically compare error distribution against a normal distribution with QQ-plots
- Apply an hypothesis test to check the normality of the errors (Kolmogorov–Smirnov, D'Agostino, etc.)

Multi-collinearity of features

$$Var(\hat{w}_j) = \frac{\sigma^2}{(m-1)Var(X_j)} \times \frac{1}{1-R_j^2}$$

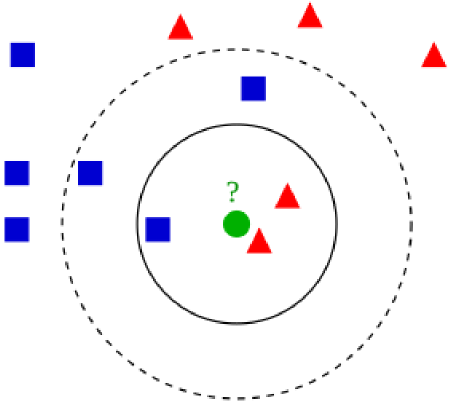
where R_j is the coefficient of determination for the linear regression explaining X_j with the remaining explanatory variables

Variance inflation factor

$$VIF_j = \frac{1}{1-R_j^2}$$

if bigger than five indicates the existence of multicollinearity.

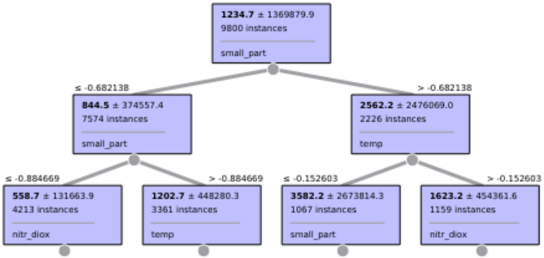
KNN K-nearest Neighbours



Main Parameters

- k : number of neighbours
- neighbour weights
- distances

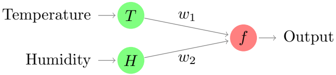
Regression tree

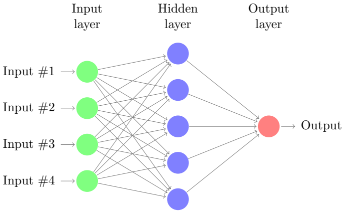


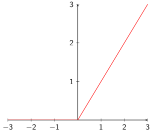
Main Parameters

- variability measure: mse (variance from mean), mae (error from median)
- max_depth
- min_samples_split: minimum number of samples to split an internal node
- min_sample_leaf: minimum number of samples required to be at a leaf node

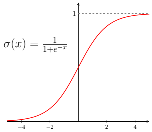
Multi-Layer Perceptron







ReLU

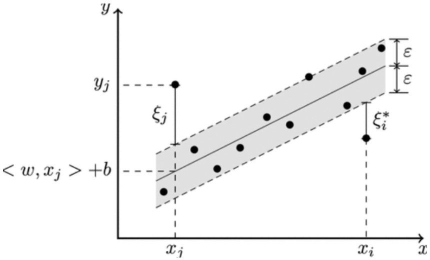


Sigmoid

Main Parameters

- hidden_layer_sizes: $(n_1, n_2, \ldots, n_L)$
- activation: identity, logistic, tanh, relu
- alpha regularization term parameter
- Resolution algorithm parameters: solver, tol, batch_size, learning_rate, max_iter.

SVR



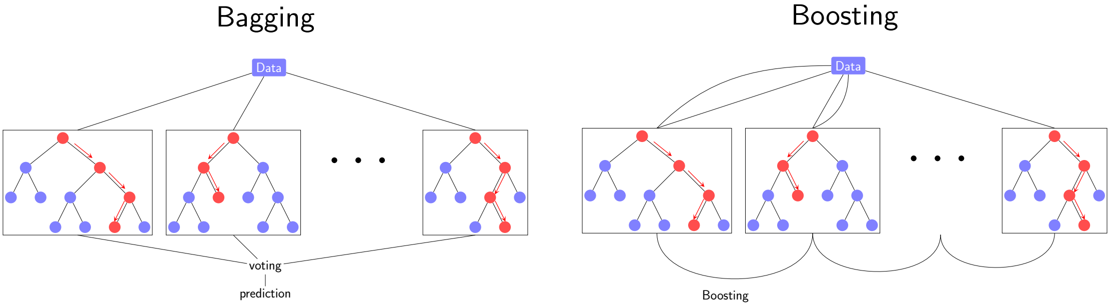
$$\min_{w,b,\zeta,\zeta^*} \frac{1}{2}\|w\| + C\sum_{i=1}^{n}(\zeta_i + \zeta_i^*)$$

$$\begin{aligned}\text{subject to } & y_i - w^T\phi(x_i) - b \leq \varepsilon + \zeta_i,\\ & w^T\phi(x_i) + b - y_i \leq \varepsilon + \zeta_i^*,\\ & \zeta_i, \zeta_i^* \geq 0, i = 1, ..., n\end{aligned}$$

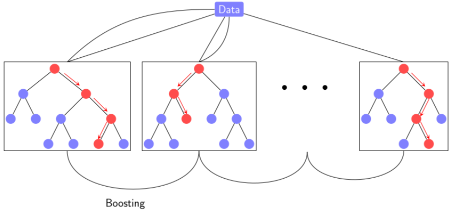
Main Parameters

- C: inverse of regularization strength
- ε: tolerance
- kernel
- Resolution algorithm parameters

Ensemble Methods



Gradient Boost



Motivation:

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \frac{f'''(a)}{3!}(x - x_0)^3 + \cdots$$

1. Train a weak learner F_0 and compute predictions $x^{(k)}$
2. For $k = 1, \ldots, K$
   - Compute the difference between the target y and the predictions of the current learner

     $$\hat{y}_{k-1} = F_{k-1}(x_i)$$

   - Train a weak learner that minimize the loss function (error)

     $$f_k = \arg\min_f L_m = \arg\min_f \sum_{i=1}^{n} l(y_i, F_{m-1}(x_i) + f(x_i))$$

   - $F_k = F_{k-1} + \lambda f_k$

Main Parameters

- n_estimators: Number of estimators (K)
- base_estimator: Weak estimator type
- learning_rate: weights of estimator in final decision (λ)