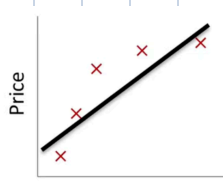


OVERFITTING & UNDERFITTING

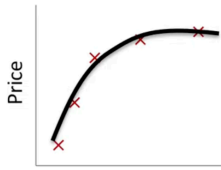
METTIAMO CASO DI DOVERE PREDIRE IL PREZZO DELLE CASE, ABBIAMO MOLTEPLICI MODELLI DOVE OGNUNO HA DEI PARAMETRI DIVERSI:



$$\theta_0 + \theta_1 x$$

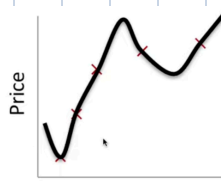
UNDERFITTING

- HIGH BIAS
- LOW VARIANCE
- LOW FLEXIBILITY



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

GIUSTO



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

OVERFITTING

- LOW BIAS
- HIGH VARIANCE
- HIGH FLEXIBILITY

SI ADATTA TROPPO AI DATI CONSIDERANDO ANCHE IL RUMORE.

BIAS BASSO: SONO FATTE MENO "IPOTESI" SUL CALCOLO DELLA FUNZIONE TARGET E QUINDI LA FUNZIONE SI ADATTA MOLTO AI DATI DI TRAINING

BIAS ALTO: LA FUNZIONE NON SI ADATTA ABBASTANZA.

COSA VOGLIAMO FARE: QUELLO CHE VOGLIAMO SCEGLIERE È IL MODELLO CHE HA MENO ERRORI SU DATI NUOVI MAI VISTI PRIMA

MODEL SELECTION & CROSS VALIDATION

function SELEZIONE-MODELLO(*AgenteCheApprende*, *esempi*, *k*) **returns** una coppia (ipotesi, tasso di errore)

err ← un array indicizzato su *dimensione*, che memorizza tassi di errore su insiemi di validazione
insieme_addestramento, *insieme_test* ← una partizione degli *esempi* in due insiemi } ①

for *dimensione* = 1 **to** ∞ **do**

err[*dimensione*] ← CONVALIDA-INCROCIATA(*AgenteCheApprende*, *dimensione*,
insieme_addestramento, *k*) } ②

if *err* sta cominciando a crescere in modo significativo **then**

dim_migliore ← il valore di *dimensione* con minimo *err*[*dimensione*] } ③

h ← *AgenteCheApprende*(*dim_migliore*, *insieme_addestramento*)

return *h*, TASSO-ERRORE(*h*, *insieme_test*) } ④

function CONVALIDA-INCROCIATA(*AgenteCheApprende*, *dimensione*, *esempi*, *k*) **returns** tasso di errore

N ← numero di esempi } ⑤

errs ← 0

for *i* = 1 **to** *k* **do**

insieme_validazione ← *esempi*[(*i* - 1) × *N*/*k* : *i* × *N*/*k*]

insieme_addestramento ← *esempi* - *insieme_validazione* } ⑥

h ← *AgenteCheApprende*(*dimensione*, *insieme_addestramento*)

errs ← *errs* + TASSO-ERRORE(*h*, *insieme_validazione*)

return *errs* / *k* // tasso di errore medio sugli insiemi di validazione nella convalida incrociata *k*-volte } ⑦

FUNZIONE **SELEZIONE - MODELLO** PRENDE 3 PARAMETRI:

- `AgenteCheApprende` : il modello o algoritmo di apprendimento.
- `esempi` : il dataset completo.
- `k` : il numero di fold per la validazione incrociata.

① INIZIALIZZAZIONE VARIABILI:

- `err` : un array che memorizza i tassi di errore calcolati per diverse "dimensioni" (gradi di complessità del modello, come profondità, numero di nodi, o gradi polinomiali).
- Il dataset `esempi` è diviso in due insiemi: `insieme_addestramento` (training set) e `insieme_test` (test set).

② CICLO PER DETERMINARE LA COMPLESSITÀ OTTIMALE DEL MODELLO

- Si itera sulle "**dimensioni**" (livelli di complessità) del modello.
- Per ogni dimensione, si calcola il tasso di errore usando la funzione `CONVALIDA-INCROCIATA`.

③ CONTROLLO TASSO DI ERRORE

- Se l'errore inizia a crescere in modo significativo (overfitting), si seleziona la dimensione migliore (`dim_migliore`), che corrisponde al **valore con il minimo errore**.
- Si addestra il modello con la dimensione ottimale (`dim_migliore`) sull'insieme di addestramento.

④ OUTPUT

- La funzione restituisce una coppia: l'ipotesi finale del modello `h` e il tasso di errore calcolato sull'insieme di test (`TASSO-ERRORE(h, insieme_test)`).

FUNZIONE **CROSS-VALIDATION**:

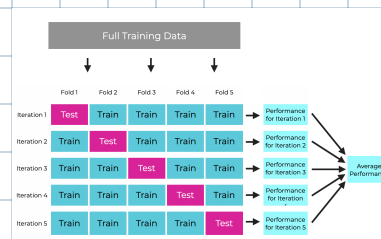
- `AgenteCheApprende` : il modello/algoritmo da testare.
- `dimensione` : il livello di complessità del modello.
- `esempi` : il dataset completo.
- `k` : il numero di fold per la validazione incrociata.

⑤ PREPARAZIONE:

- Si determina `N`, il numero totale di esempi nel dataset.
- Si inizializza `errs`, una variabile per accumulare gli errori.

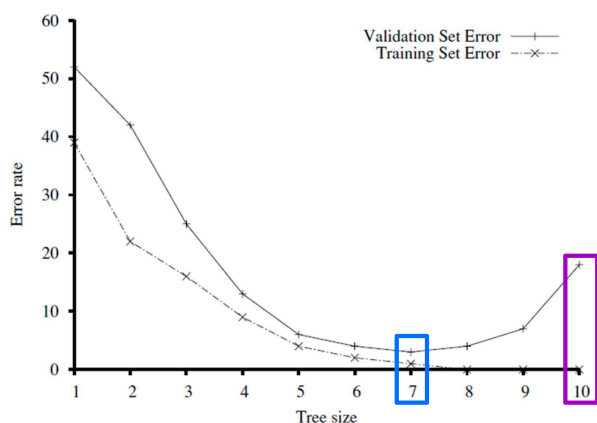
⑥ CICLO SUI FOLD

- Si divide il dataset in:
 - `insieme_validazione` : una partizione usata per il test.
 - `insieme_addestramento` : la restante parte usata per l'addestramento.
- Si addestra il modello sull'insieme di addestramento e si **calcola il tasso di errore sull'insieme di validazione**.
- Il tasso di errore viene accumulato in `errs`.



7 OUTPUT

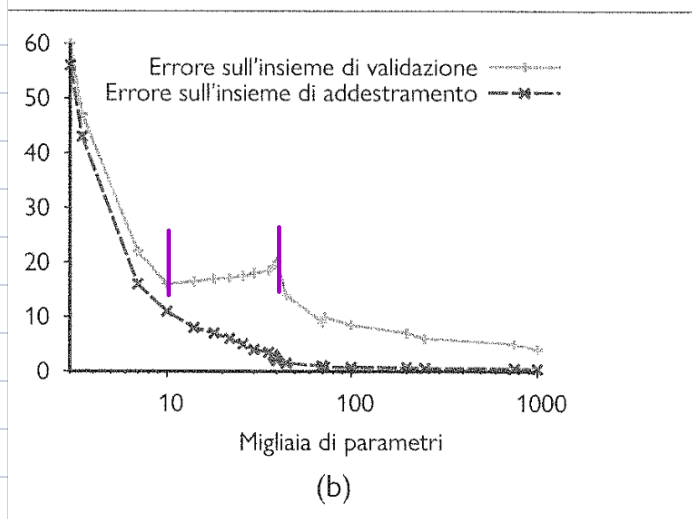
- La funzione restituisce il tasso di errore medio, calcolato dividendo `errs` per `k`.



ESEMPIO CON ALBERO DI DECISIONE

CI FERMIAMO QUANDO COMINCIA A CREARSI **OVERFITTING** E PRENDIAMO L'ALBERO CON L'ERRORE MINIMO, IN QUESTO CASO 7.

QUESTO È IL MIGLIOR COMPROMESSO TRA OVERFITTING E UNDERFITTING.



ESEMPIO CON RETI NEURALI

LE LINEE VIOLA INDICANO UN PUNTO IN CUI LA RETE NON HA PERFORMATO BENE, INVECE DI FERMARCI APPENA NOTIAMO CHE IL VALID. SET PEGGIORA, ABBIAMO UNA "FINESTRA DI PAZIENZA" CHE USIAMO PER VEDERE SE POI IL TASSO DI ERRORE SI ABBASSA DI NUOVO.

APPENDIMENTO

OUTPUT ATESO È :
 → UN ELEMENTO DI UN INSIEME FINITO (es. VOF) = PROBLEMA DI **CLASSIFICAZIONE**
 → UN NUMERO (es. TEMPERATURA) = PROBLEMA DI **REGRESSIONE**

TIPI DI APPENDIMENTO

SUPERVISIONATO = COPPIE DI INPUT/OUTPUT, L'AGENTE APPRENDE UNA FUNZIONE CHE FA CORRISPONDERE L'INPUT ALL'OUTPUT.

NON SUPERVISIONATO = L'AGENTE APPRENDE PATTERN NELL'INPUT SENZA ALCUN FEEDBACK. (es. CLUSTERING)

PER RINFORZO = RINFORZO O VERO RICOMPENSA O PUNIZIONE, IN BASE AL RINFORZO RICEVUTO L'AGENTE DECIDE QUALI OPERAZIONI PRECEDENTI HANNO PORTATO A QUEL RINFORZO.

APPENDIMENTO SUPERVISIONATO

DATO UN INSIEME DI ADDESTAMENTO, COSTITUITO DA N COPPIE DI ESEMPLI DI INPUT & OUTPUT, OGNI COPPIA È STATA GENERATA DA UNA FUNZIONE $y = f(x)$, IL NOSTRO OBIETTIVO È SCOPRIRE UNA FUNZIONE h , CHIAMATA IPOTESI, CHE APPROSSIMI f . h FA PARTE DELLO SPAZIO DELLE IPOTESI.

APPROCCIO DISCRIMINATIVO = METODI CHE CERCANO DI MODELLARE UNA **FRONTIERA DI DECISIONE** TRA LE DIVERSE CLASSI DI DATI, CIOÈ CERCANO DI TROVARE UNA FUNZIONE CHE SEPARI AL MEGLIO I DATI IN BASE ALE LORO **ETICHETTE**.
UTILI IN PROBLEMI DI CLASSIFICAZIONE.

ESEMPIO
DI METODO
↓

METODO LINEARE = NEGLI APP. DISCR. I METODI LINEARI USANO DELLE FUNZIONI LINEARI PER DEFINIRE LA **FRONTIERA DI DECISIONE**.

↓
FORMULA DELLA FUNZ. LINEARE: $h(x) = \text{SIGN}(W \cdot x + b)$ CON:

- $h(x)$ = FUNZ. CHE ASSEGNA L'ETICHETTA
- W = VETTORE DI **PESI** CHE DETERMINA L'INCLINAZIONE
- x = VETT. DI **DATI IN INPUT**
- b = **BIAS**, PER SPOSTARE LA FRONT.

UN ALTRO ESEMPIO DI MODELLO NELL' APP. SUPERVISIONATO SONO GLI **ALBERI DI DECISIONE**, UTILIZZATI AD ESEMPIO IN PROBLEMI COME QUELLO DELL' ATTESA AL RISTORANTE.

Riassunto delle Differenze e Ambiti di Utilizzo

Metodo	Descrizione	Quando utilizzarlo	Esempi di applicazione
Approccio Discriminativo	Modella direttamente la probabilità di una classe data l'input	Quando si vuole distinguere chiaramente le classi	Classificazione di email (spam/non-spam), riconoscimento immagini
Metodo Lineare	Modello che assume una relazione lineare tra input e output	Quando i dati sono linearmente separabili o la relazione è approssimativamente lineare	Previsione prezzi case (regressione lineare), analisi rischio credito
Alberi di Decisione	Modello che suddivide i dati in base a condizioni sui valori delle caratteristiche	Quando la relazione tra le caratteristiche e l'etichetta è non lineare o complessa	Diagnosi mediche, approvazione prestiti, analisi di rischio