

4. How to evaluate the hypothesis

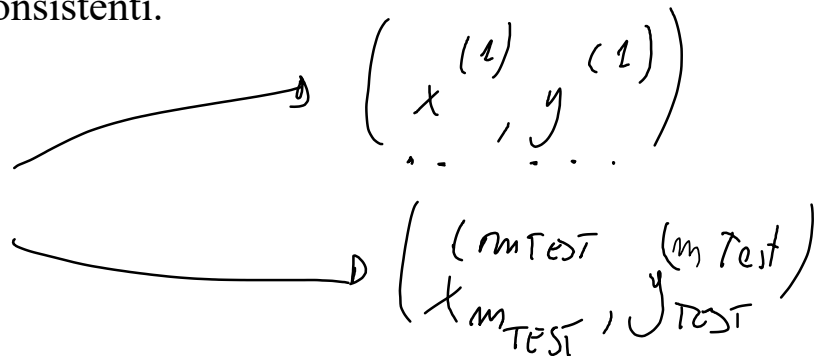
martedì 29 ottobre 2024 11:21

L'obiettivo è creare un modello che generalizzi bene sui dati nuovi, ottenendo prestazioni affidabili e consistenti.

N.B

Il training set vale 80%

Il TEST SET vale 20%



TRAINING / TESTING PROCEDURE

Addestramento del Modello (Training):

- Nella regressione lineare, l'obiettivo è trovare i parametri (coefficienti) θ che minimizzano l'errore sui dati di addestramento.

Per calcolare il test set error dobbiamo utilizzare tale formula:

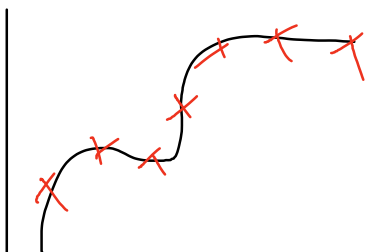
$$J(\theta) = \frac{1}{2 m_{\text{TEST}}} \sum_{i=1}^m \left(h_{\theta}(x^{(i)}) - y^{(i)} \right)^2$$

TRAINING/TESTING FOR LOGISTIC REGRESSION

Nella regressione logistica, l'obiettivo è apprendere i parametri (coefficienti) θ che permettono di separare al meglio le classi, minimizzando l'errore di classificazione sui dati di addestramento.

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \left(y^{(i)} \log h_{\theta}(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_{\theta}(x^{(i)})) \right)$$

THE GENERALIZATION ERROR IN PRACTICE / ERRORE DI GENERALIZZAZIONE



L'overfitting si verifica quando un modello è eccessivamente adattato ai dati di addestramento. In questo caso, il modello segue troppo da vicino i punti dati di addestramento

Un modello overfittato avrà un errore molto basso sul set di addestramento perché segue perfettamente i dati su cui è stato addestrato.

- Tuttavia, questo modello potrebbe non generalizzare bene quando viene applicato a nuovi dati (mai visti durante l'addestramento), perché cattura rumori o dettagli specifici del set di addestramento che non rappresentano il comportamento generale del fenomeno.

MODEL SELECTION | QUALE MODELLO UTILIZZARE

L'obiettivo è confrontare diverse ipotesi (ossia modelli) variando il grado del polinomio utilizzato nella funzione $h_{\theta}(x)$

D1 = grado del polinomio in generale Dn

$$\begin{aligned} q_1 &= h_{\theta}(x) = \theta_0 + \theta_1 x && \longrightarrow J_{\text{TEST}}(\theta^{(1)}) \\ q_2 &= h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 && \longrightarrow J_{\text{TEST}}(\theta^{(2)}) \\ &\vdots \\ q_m &= h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \dots + \theta_m x_m && \dots \end{aligned}$$

Per ogni grado di polinomio, si ottiene un set di parametri θ ($\theta^{(1)}, \theta^{(2)}, \theta^{(3)}, \dots$) e si calcola l'errore o costo sul set di test, indicato come $J(\theta)$

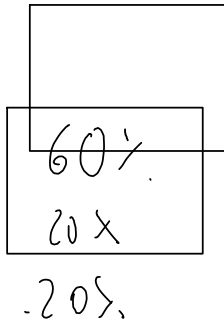
- Questo permette di identificare quale grado di polinomio porta al minore errore sul set di test.

3. Scelta del modello migliore:

- Immaginando che il polinomio di grado 5 mostri il valore più basso della funzione di costo $J(\theta^{(5)})$, verrebbe scelto come modello migliore su quel set di test.

CROSS-VALIDATION

La cross-validation è una tecnica di validazione dei modelli di machine learning utilizzata per stimare le loro prestazioni in modo più accurato. Consiste nel suddividere il set di dati in più parti o fold, che vengono utilizzate alternativamente per l'addestramento e per il test. Questo approccio aiuta a garantire che il modello non sia valutato solo su una singola suddivisione dei dati, ma su più partizioni, permettendo una stima più robusta e affidabile delle prestazioni.



Cross-Validation Set: Questo set viene utilizzato per ottimizzare e validare il modello. Serve a testare le prestazioni del modello durante l'allenamento, ma senza incidere sul modello finale. In particolare, è usato per:

Valutare l'efficacia delle impostazioni dei parametri (hyperparameters tuning).


Evitare il sovradattamento (overfitting), cioè quando il modello funziona bene sul training set ma male su nuovi dati.

DOBBIAMO SCEGLIERE IL MODELLO CHE PERFORMA MEGLIO ALL'INTERNO DEL VALIDATION SET

La cross validation (validazione incrociata) è una tecnica usata per valutare le prestazioni di un modello di machine learning, aiutando a trovare la miglior configurazione e prevenire il problema del sovradattamento (overfitting), cioè quando il modello si adatta troppo bene ai dati di addestramento ma ha difficoltà a generalizzare su dati nuovi.

Ecco i concetti principali: 

Hold-out Cross Validation:

In questo approccio, si divide il dataset di addestramento in due parti: una per addestrare il modello e l'altra (tipicamente 1/3 del set di addestramento) per valutarlo. Questo set separato per la valutazione viene chiamato validation set. Lo svantaggio è che il modello non usa tutto il set di addestramento per l'apprendimento e, quindi, può perdere informazioni utili. 

K-Folds Cross Validation:

In questo metodo, il dataset di addestramento è diviso in k sottoinsiemi (o "folds"). Si addestra il modello k volte, ogni

volta usando k-1 fold per l'addestramento e il fold rimanente per la validazione.

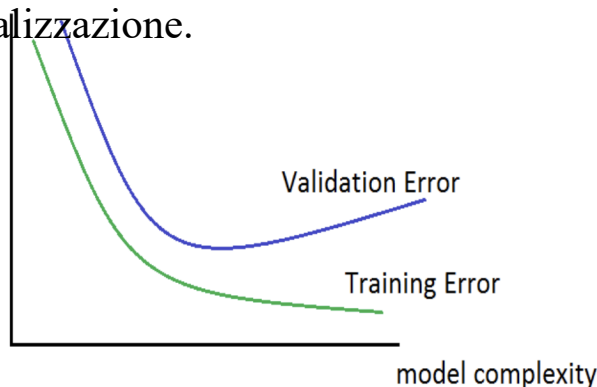
Si calcola la media delle prestazioni ottenute in ogni iterazione per ottenere una stima più affidabile delle prestazioni del modello. Questo approccio è ~~più~~ robusto rispetto all'hold-out, poiché ogni dato viene usato sia per l'addestramento che per la validazione.

Utilizzo nella selezione degli Iperparametri:

Gli iperparametri sono parametri che influenzano il processo di apprendimento (es., il tasso di regolarizzazione). La cross validation aiuta a testare varie configurazioni e scegliere quella che porta alle migliori prestazioni, evitando che il modello si adatti troppo o troppo poco ai dati.

Selezione delle Caratteristiche:

La cross validation può anche essere impiegata per scegliere le caratteristiche (o features) più rilevanti. In ogni fold si prova un sottoinsieme diverso di caratteristiche e si sceglie il set che ottimizza la prestazione del modello, riducendo anche la complessità del modello e migliorando la generalizzazione.



Questo grafico mostra come cambiano gli errori di training (addestramento) e di validation (validazione) in funzione della complessità del modello.

Training Error (errore di addestramento)**: È l'errore che il modello commette sui dati di addestramento. All'aumentare della complessità del modello (ad esempio aggiungendo più parametri o nodi in una rete neurale), il modello riesce ad adattarsi meglio ai dati di addestramento, riducendo l'errore. Per questo motivo, la

curva dell'errore di addestramento diminuisce man mano che aumenta la complessità del modello.

Validation Error (errore di validazione): Questo è l'errore che il modello commette su un set di dati separato, utilizzato per valutare la capacità di generalizzazione del modello. All'inizio, aumentando la complessità del modello, anche l'errore di validazione diminuisce, poiché il modello riesce a catturare meglio i pattern presenti nei dati. Tuttavia, superato un certo punto, la complessità aggiuntiva fa sì che il modello si adatti troppo ai dati di addestramento (overfitting), catturando anche il rumore e le variazioni specifiche del dataset di addestramento. Questo fa aumentare l'errore di validazione, poiché il modello non è più in grado di generalizzare bene.

Dopo aver scelto un modello, è necessario valutare le sue prestazioni su un set di test per capire quanto bene il modello generalizzi su nuovi dati. Questo processo segue alcune regole importanti:

Uso di un Test Set: Per una valutazione finale del modello, bisogna usare un test set, che non è stato coinvolto né nell'addestramento né nella validazione. Questo set contiene dati "nuovi" rispetto a quelli visti durante la selezione del modello.

Motivazione per il Test Set: Valutare il modello su un test set serve a verificare se il modello generalizza abbastanza bene, ovvero se può fare previsioni accurate su dati che non ha mai visto prima.

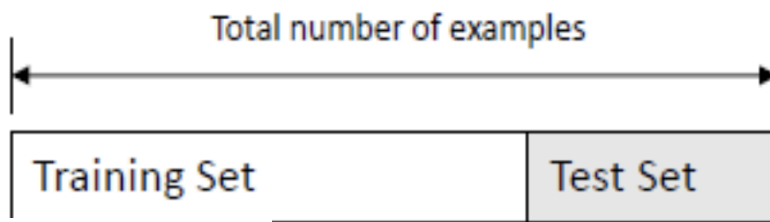
Corretta Separazione dei Dati: È scorretto usare il test set durante la selezione del modello, perché rischierebbe di far scegliere un modello che si adatta al test set anziché generalizzare. Il test set deve essere usato solo quando il modello è stato scelto e si desidera una valutazione finale.

Metodi per Dividere il Dataset: Esistono due principali metodi per dividere i dati in set di addestramento e test:

Holdout: Si divide una parte dei dati per l'addestramento e una parte separata per il test.

K-fold Cross Validation: Si divide il dataset in k parti e si ripete l'addestramento k volte, usando ogni volta una parte diversa come set di validazione e le altre $k-1$ parti per l'addestramento.

+



HOLDOUT

Nella tecnica Holdout, il dataset viene diviso in due insiemi principali: Training Set e Test Set. Ecco i dettagli:

Problema della Rappresentatività: La scelta della divisione tra training set e test set può influenzare le prestazioni del modello. Ad esempio, se una classe è rappresentata in modo molto diverso tra i due set, il modello potrebbe non riuscire a generalizzare correttamente.

Soluzione - Stratificazione: Per risolvere il problema di distribuzione delle classi, si può usare la stratificazione. Con la stratificazione, il dataset viene diviso in modo che ogni classe sia rappresentata proporzionalmente sia nel training set sia nel test set. Questo migliora l'affidabilità delle prestazioni del modello, poiché entrambi i set riflettono la distribuzione reale delle classi nel dataset originale.

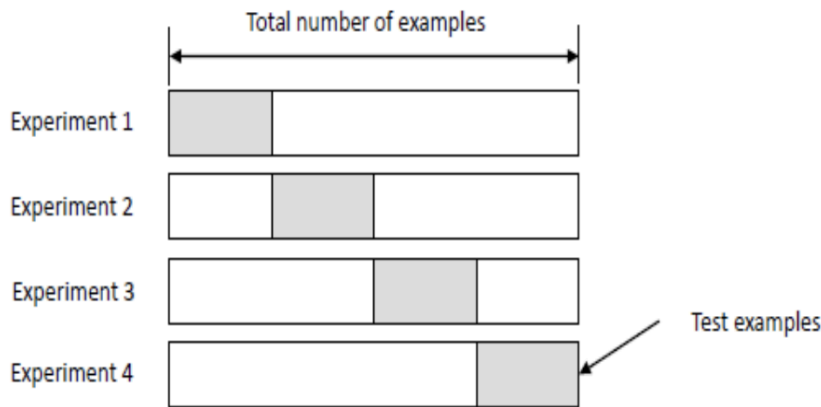
LIMITATION OF USING SINGLE TRAINING/TEST

Dati insufficienti: Se i dati sono pochi, dividerli può creare set troppo piccoli, rendendo meno affidabile la stima delle prestazioni.

2. **Set di test ridotto:** Un test set piccolo può dare stime meno stabili dell'accuratezza.

3. **Set di training ridotto:** Meno dati per l'addestramento significano che il modello potrebbe non apprendere bene le caratteristiche dei dati.

4. **Sensibilità ai campioni:** Con una sola divisione, non possiamo vedere quanto l'accuratezza cambi con dati di training diversi.



Cos'è la K-fold Cross Validation?

L'insieme di dati originale viene diviso in

k sottoinsiemi di dimensioni uguali, chiamati fold.

Ogni volta, uno di questi k fold viene utilizzato come set di test mentre gli altri

$k-1$ fold vengono utilizzati come set di addestramento.

Questo processo viene ripetuto k volte, così ogni fold viene utilizzato esattamente una volta come set di test.

I risultati ottenuti dai k test vengono mediati per ottenere una singola stima delle prestazioni del modello.

Vantaggi della K-fold Cross Validation

Utilizzo completo dei dati: tutte le osservazioni sono usate sia per il training che per il test.

Singolo utilizzo per il test: ogni osservazione viene usata una sola volta come set di test, riducendo il rischio di bias di valutazione.



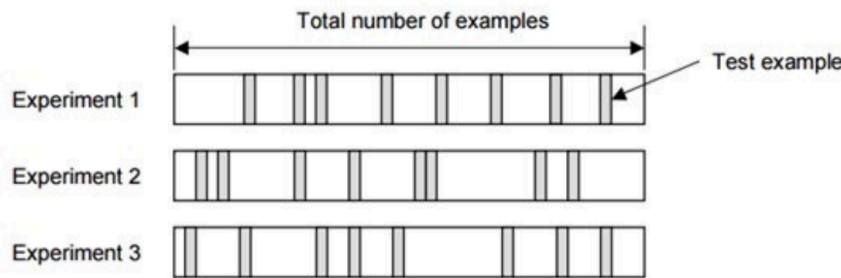
+ RANDOM SUBSAMPLING

Il Random Subsampling è una tecnica di validazione machine learning per stimare la precisione di un modello su dati non visti. Consiste nel dividere casualmente il dataset in due sottogruppi: un set di addestramento e un set di test, ripetendo il processo più volte per ottenere una stima più accurata e stabile delle prestazioni del modello.

OBIETTIVO: affrontare il problema di ottenere una stima più affidabile delle prestazioni di un modello, ripartendo casualmente il dataset in un set di addestramento e uno di test più volte.

Viene eseguita una suddivisione casuale del dataset (o split), in cui vengono selezionati casualmente un numero fisso di esempi (senza ripetizione) per i set di addestramento e di test.

- Ogni suddivisione rappresenta un esperimento separato in cui il modello viene addestrato con i dati del set di training e valutato con il set di test.
- Questa procedura viene ripetuta per K volte (il numero di ripetizioni è deciso dall'utente), e ogni volta si generano nuovi set di addestramento e test.



L'immagine mostra più esperimenti (1, 2 e 3) dove ogni esperimento rappresenta una suddivisione diversa del dataset in set di addestramento e test.

- In ciascun esperimento, alcuni esempi del dataset sono assegnati al test set (mostrati come aree vuote), mentre gli altri esempi fanno parte del training set (aree riempite).

IL PROCESSO E' COSI ESEGUITO:

Per ogni suddivisione, il modello viene addestrato da zero con il training set e testato sul test set per calcolare l'errore.

- Alla fine, gli errori o le metriche di performance calcolati per ogni suddivisione vengono mediati per ottenere una stima complessiva delle prestazioni del modello.

Vantaggi del Random Subsampling

- Fornisce una stima più robusta delle prestazioni del modello rispetto a una singola suddivisione del dataset.
- Permette di valutare la stabilità del modello su diverse suddivisioni casuali del dataset.

Svantaggi

- Poiché alcuni campioni potrebbero essere selezionati più volte nei set di training e test, il metodo può risultare ridondante.
- Non garantisce che ogni campione venga utilizzato esattamente una volta nei set di test, il che può portare a una valutazione meno uniforme rispetto alla k-fold cross-validation.

ERROR ESTIMATE K-FOLDS

Case i	Train on					Test on	Error
Case1		F2	F3	F4	F5	F1	1.5
Case2	F1		F3	F4	F5	F2	0.5
Case3	F1	F2		F4	F5	F3	0.3
Case4	F1	F2	F3		F5	F4	0.9
Case5	F1	F2	F3	F4		F5	1.1

$$Error = \frac{1}{k} \sum_{i=1}^k Error_i$$

. Suddivisione del Dataset:

- Il dataset viene diviso in K sottoinsiemi, o “folds”. In questo esempio specifico, ogni caso rappresenta uno di questi fold.
- A turno, ogni fold viene utilizzato come set di test, mentre i restanti K-1 fold vengono usati come set di addestramento.

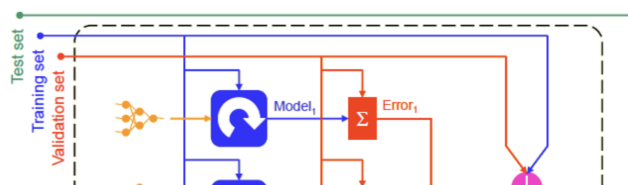
Una volta calcolato l'errore per ogni fold, si fa una media di tutti questi errori per ottenere una stima complessiva dell'errore del modello.

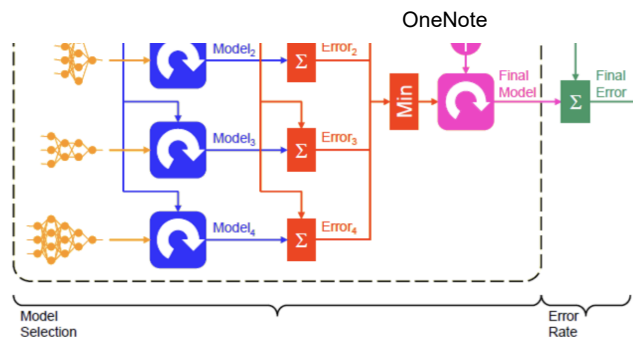
Vantaggi della K-fold Cross-Validation

- Fornisce una stima robusta delle prestazioni del modello su dati non visti.
- Utilizza tutto il dataset sia per l'addestramento che per il test, aumentando l'affidabilità della valutazione.

Per quanto riguarda lo svantaggio fondamentale che per dataset grandi è computazionalmente costoso.

Cross-Validation overview





CICLO DI VITA DI UN MODELLO MACHINE LEARNING



ML system life cycle (revisited)

