

MACHINE LEARNING

Module 2

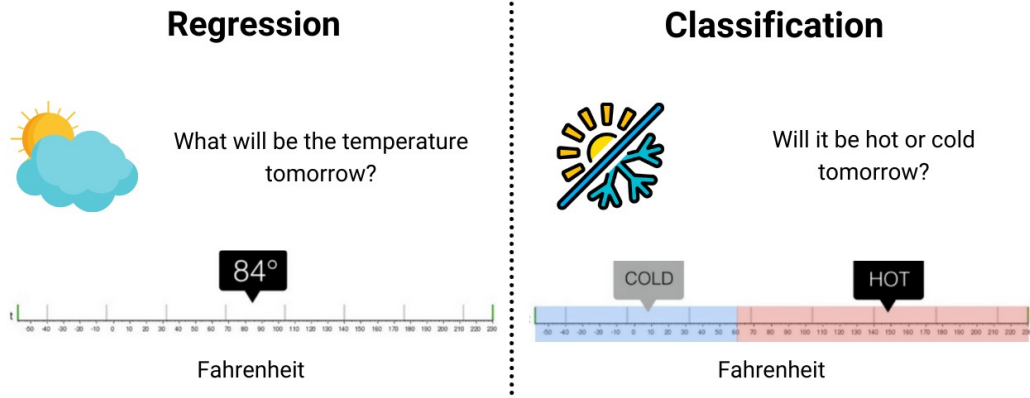
Machine Learning Models *(a gentle introduction)*

Carrellata di Modelli di Machine Learning.

Module 2: Classification VS Regression

Regression: involves the estimation of a continuous value as output

Classification: involves the assignment of a label or category to a given input



Regolazione (Regression):

- *Definizione:* La regressione prevede la stima di un valore continuo come output. Si adatta bene quando si tratta di quantità continue
- *Esempio:* Predire il prezzo di una casa in base alle sue caratteristiche, è ideale per problemi di assegnazione a categorie discrete..

Classificazione:

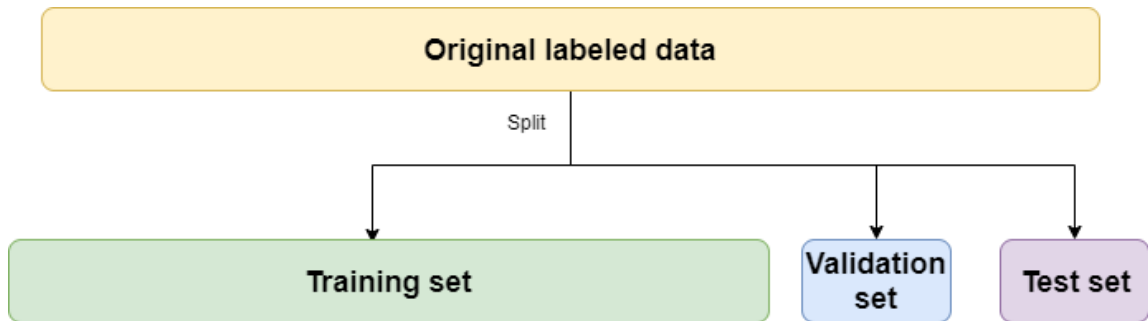
- *Definizione:* La classificazione prevede l'assegnazione di un'etichetta o categoria a un dato input.
- *Esempio:* Classificare le email come spam o non spam.
- **Esempi Aggiuntivi:**
 - Altri esempi di task di regressione includono la previsione della temperatura o il reddito annuo.
 - Altri esempi di task di classificazione includono la diagnosi medica (malato/sano) o il riconoscimento di oggetti in un'immagine.
- La scelta tra regressione e classificazione dipende dalla natura del problema e dal tipo di output desiderato.

Module 2: Dataset Splitting

Train: labelled examples on which the model will learn the relationships between input and output

Validation: used during training to adjust the hyper-parameters of the model without affecting the test set

Test: new and unseen data that the model must generalise

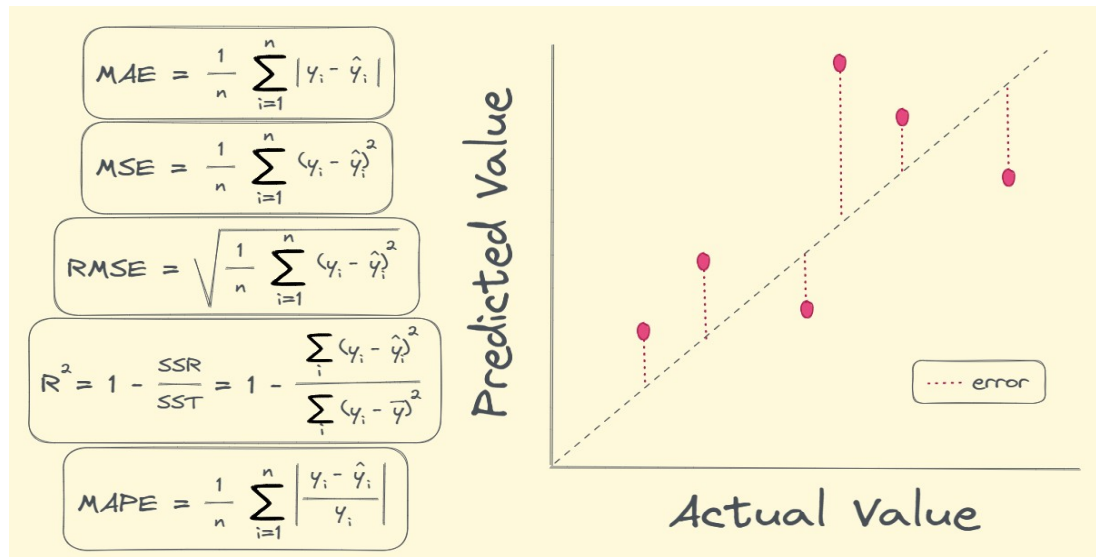


Train Set: Il set di addestramento è utilizzato per addestrare il modello. Contiene esempi etichettati su cui il modello imparerà le relazioni tra input e output.

Test Set: Il set di test è utilizzato per valutare le prestazioni del modello dopo l'addestramento. Contiene dati nuovi e non visti che il modello deve generalizzare.

Validation Set: Il set di convalida è utilizzato durante l'addestramento per regolare gli iperparametri del modello senza influenzare il set di test. Aiuta a evitare l'overfitting.

Module 2: Evaluation Metrics for *Regression*



MAE (Mean Absolute Error): Calcola la media delle differenze assolute tra le previsioni del modello e i valori effettivi. È robusto agli outliers.

Scenario: State lavorando su un'applicazione in cui gli errori nel prezzo previsto devono essere interpretati in modo lineare e dove errori più grandi e più piccoli hanno la stessa importanza.

- *Esempio:* Se un modello prevede il prezzo di un'abitazione con un errore medio di 10.000 dollari, questo valore avrà un impatto diretto e facilmente comprensibile sull'esperienza dell'utente o sulle decisioni aziendali.

MSE (Mean Squared Error) e RMSE (Root Mean Squared Error): Calcola la media delle differenze quadratiche tra le previsioni del modello e i valori effettivi. Gli errori più grandi sono maggiormente penalizzati.

- *Scenario:* Se stai lavorando in un contesto in cui gli errori più grandi dovrebbero essere **penalizzati** in modo significativo rispetto agli errori più piccoli, poiché potrebbero indicare una predizione meno accurata.
- *Esempio:* Se un modello ha un MSE di 100 milioni, questo

Module 2: Evaluation Metrics for *Classification*

Matrice di Confusione:

- TP (True Positive), TN (True Negative)
- FP (False Positive); FN (False Negative)

Accuracy:
$$\frac{TP+TN}{TP+TN+FP+FN}$$

Precision
$$\frac{TP}{TP+FP}$$

Recall
$$\frac{TP}{TP+FN}$$

F1 Score
$$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$



Matrice di Confusione:

- TP (True Positive): Numero di osservazioni correttamente classificate come positive.
- TN (True Negative): Numero di osservazioni correttamente classificate come negative.
- FP (False Positive): Numero di osservazioni negative erroneamente classificate come positive (Errore di Tipo I).
- FN (False Negative): Numero di osservazioni positive erroneamente classificate come negative (Errore di Tipo II).

Accuracy (Accuratezza): Calcola la percentuale di previsioni corrette rispetto al totale delle previsioni cioè Numero di previsioni corrette diviso il totale delle previsioni.

Precision (Precisione): misura la percentuale di previsioni positive corrette tra tutte le previsioni positive.

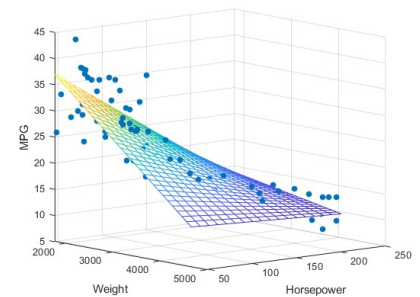
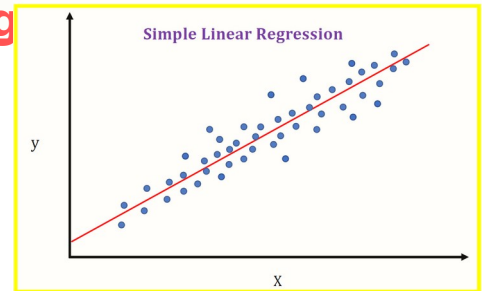
Recall (Recupero o Sensibilità): La percentuale di previsioni positive corrette tra tutte le reali positività.

F1 Score: Combina precision e recall (Media armonica) in un'unica metrica bilanciata.

È importante scegliere le metriche in base alle caratteristiche specifiche del problema e ai requisiti del caso d'uso. Ad esempio, in

Module 2: Intro to Linear Reg

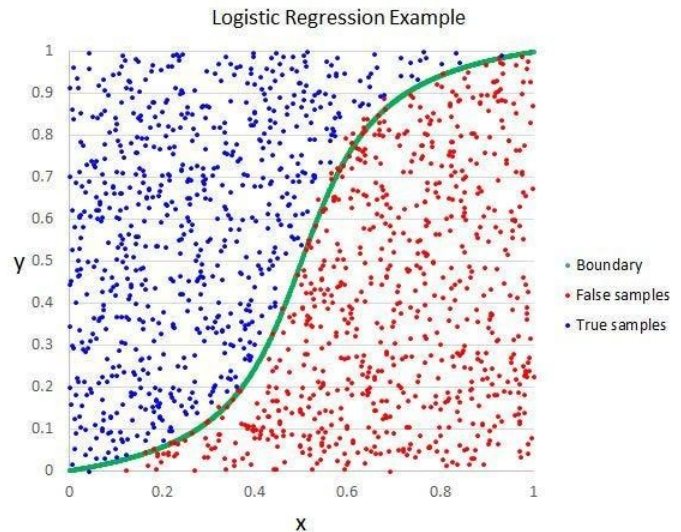
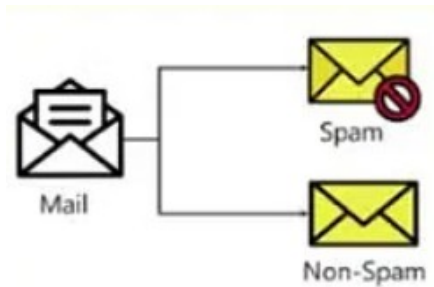
- Linear regression determine the strength and relationship of two or more variables.
- Used to **predict** the value of a dependent variable based on the value of one or more independent variables.
- two types of linear regression:
 - *Simple* Linear Regression (one independent variable) and
 - *Multiple* Linear Regression (more than one independent variable)



- La regressione lineare è una metodologia statistica che ci permette di determinare la forza e la relazione tra due o più variabili. In altre parole, ci aiuta a capire come le variabili sono correlate tra loro.
- Viene utilizzata per prevedere il valore di una variabile dipendente (cioè il risultato che vogliamo prevedere) in base al valore di una o più variabili indipendenti (cioè le informazioni che usiamo per fare la previsione).
- Esistono due tipi di regressione lineare: la Regressione Lineare Semplice (quando abbiamo una sola variabile indipendente) e la Regressione Lineare Multipla (quando abbiamo più di una variabile indipendente).
- Esempio Regressione Lineare Semplice: una volta capito come le ore di studio influenzano il voto dell'esame, potremmo prevedere quale voto otterremmo se studiassimo un certo numero di ore. Regressione Lineare Multipla: potremmo voler capire come il voto dell'esame è influenzato sia dalle ore di studio che dal numero di libri letti.

Module 1: Logistic Regression (... for Classification)

$$P = \frac{e^{a+bX}}{1 + e^{a+bX}}$$



La regressione logistica è un modello statistico utilizzato per problemi di **classificazione**.

A differenza della regressione lineare, che è utilizzata per prevedere valori numerici, la regressione logistica è progettata per prevedere la **probabilità** che un'osservazione appartenga a una determinata classe.

Ad esempio, può essere utilizzata per prevedere se un'email è spam o non spam, se un paziente ha una determinata malattia o no, o se un cliente acquisterà un prodotto o meno.

Il modello di regressione logistica **utilizza la funzione logistica** per trasformare la variabile dipendente in modo che cada nell'intervallo tra 0 e 1, che può essere interpretato come una probabilità. Questa trasformazione consente di modellare e prevedere le probabilità di appartenenza a una classe piuttosto che prevedere valori numerici.

In sintesi, la regressione logistica è un modello statistico utilizzato per prevedere le probabilità di appartenenza a una classe e viene comunemente utilizzato in problemi di classificazione binaria.

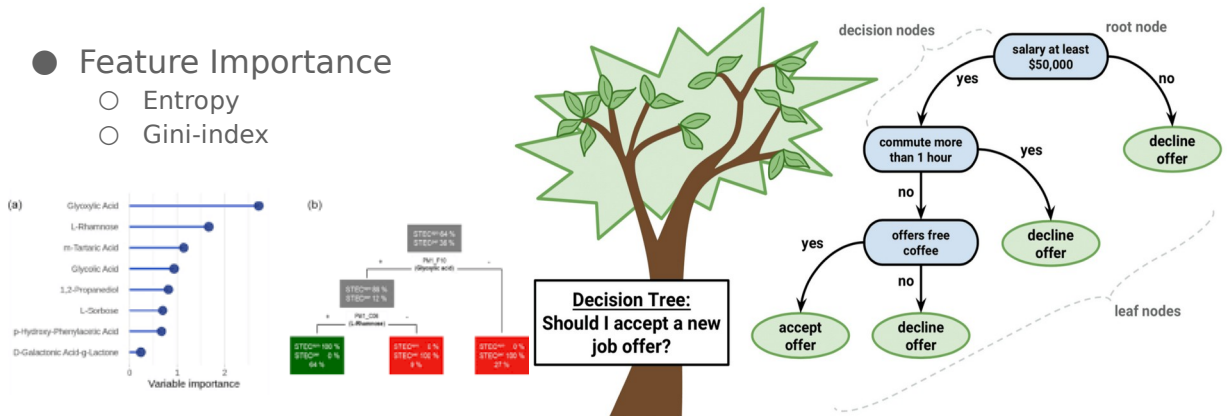
Module 2: Decision Tree (DT) & Feature Importance

● Classification & Regression

- Explainable AI

● Feature Importance

- Entropy
- Gini-index



I decision tree sono modelli di apprendimento automatico utilizzati per prendere decisioni. Funzionano in modo simile a un flusso decisionale in cui vengono poste domande su determinate variabili e, in base alle risposte, si procede lungo i rami dell'albero fino a giungere a una decisione.

I decision tree sono utilizzati in diversi campi, come l'analisi decisionale, la classificazione e la regressione.

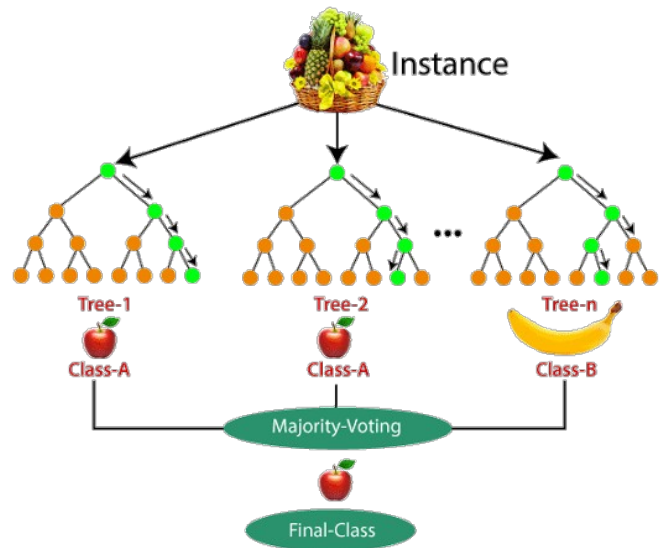
Sono facili da interpretare e possono gestire sia dati categorici che numerici. Possono essere visualizzati (interpretati visivamente) e compresi facilmente. Ciò li rende utili in ambiti come medico e/o legale.

I decision tree sono composti da nodi decisionali e nodi foglia, e vengono costruiti in base alle variabili che meglio separano i dati in classi omogenee. Sono utili per comprendere quali variabili siano importanti per prendere una decisione e come queste variabili siano collegate tra loro.

La feature importance, si riferisce alla valutazione dell'influenza di ciascuna variabile (o caratteristica) all'interno di un modello di machine learning. Nei modelli di albero decisionale, come i decision tree, le feature importance sono calcolate in base a come le variabili vengono utilizzate per suddividere i dati durante la costruzione dell'albero. Le variabili che sono più spesso utilizzate per le suddivisioni, e che meglio separano i dati in classi omogenee, vengono considerate più importanti per la previsione del modello. L'entropia e l'indice di Gini sono due misure utilizzate nei decision tree per valutare la feature importance, cioè l'importanza delle

Module 2: Random Forest (Classification & Regression)

- Ensemble
- Mean Prediction
- Robust to Overfitting



Il Random Forest è un algoritmo di machine learning che combina l'output di più alberi decisionali in un unico modello. Questo approccio ensemble lo rende adatto sia per compiti di classificazione che di regressione. In pratica, il Random Forest addestra ciascun albero su un **insieme leggermente diverso** di osservazioni, dividendo i nodi in ogni albero considerando un numero limitato di caratteristiche. Le previsioni finali del Random Forest sono fatte facendo la media delle previsioni di ogni singolo albero, il che lo rende più robusto e meno soggetto all'overfitting rispetto a un singolo albero decisionale.

Questo algoritmo è ampiamente utilizzato in diversi settori per prendere decisioni di business migliori, grazie alla sua capacità di gestire sia dati categorici che numerici e di gestire automaticamente i valori mancanti.

Module 2: Gradient Boosting Machine (GBM)

● Weak models to a Strong Model

- loss function
- actual value y_i , current prediction $F(x_i)$

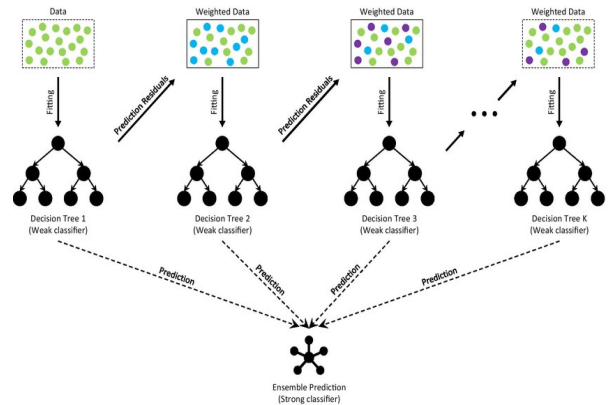
● Gradient

$$F_t(x)$$

- current model
- new model
- New & Current model combination

$$F_{t+1}(x) = F_t(x) + \eta h_{t+1}(x)$$

- iteration to block criterion



Si basa sull'idea di combinare diversi modelli deboli in un unico modello forte.

Durante l'addestramento, ogni nuovo modello cerca di correggere gli errori del modello precedente, concentrandosi sulle istanze che sono state predette in modo errato.

In pratica, il modello calcola il gradiente della funzione di perdita rispetto alle previsioni correnti e addestra un nuovo modello per minimizzare questo gradiente.

Questo processo viene ripetuto per un numero specificato di iterazioni, consentendo al modello di apprendere e migliorare progressivamente.

Il modello aggiorna le previsioni in direzione **opposta** al gradiente della funzione di perdita. Il GBM minimizza una loss funzione attraverso il *gradiente* della funzione di perdita.

La funzione di perdita può variare a seconda del problema, ma comunemente si utilizza la perdita quadratico o la perdita di Huber.

Definiamo il modello corrente: $F_t(x)$ dopo t iterazioni.

Indichiamo la funzione di perdita come rispetto alle previsioni correnti:

y_i è il valore reale della variabile di risposta per l'osservazione

$F(x_i)$ è la predizione corrente del modello per l'osservazione i .

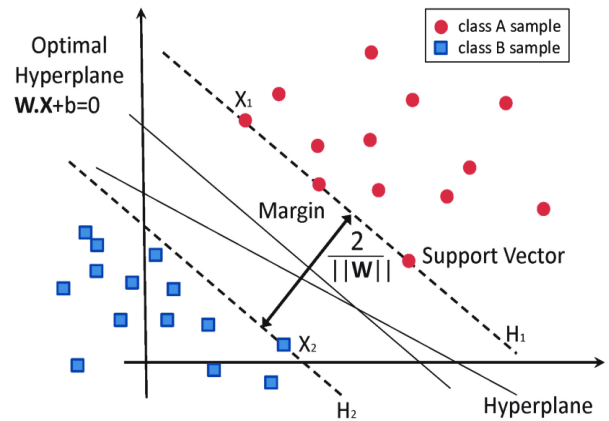
Module 2: Support Vector Machine (SVM)

- Hyperplan Separation (Classification)
- Hyperplan Approximation (Regression)

$$L(y, f(x)) = \max(0, 1 - y \cdot f(x))$$

● LOSS FUNCTION GRADIENT

$$\bullet \min_{w,b} \frac{1}{2} ||w||^2 + C \sum_{i=1}^n L(y_i, f(x_i))$$



La sua idea principale è trovare il piano (o l'iperpiano) che massimizza la separazione tra le classi nel caso di classificazione, o che approssima al meglio la relazione tra le variabili indipendenti e dipendenti nel caso di regressione.

Per SVM, comunemente si utilizza la funzione di perdita hinge per la classificazione. La funzione di perdita hinge per una singola osservazione è definita come:

$$L(y, f(x)) = \max(0, 1 - y \cdot f(x))$$

Dove: y è l'etichetta della classe (1 o -1), $f(x)$ è la funzione di decisione che rappresenta la distanza di un punto x dal piano decisionale.

L'obiettivo è minimizzare la somma delle funzioni di perdita per tutte le osservazioni, penalizzando gli errori di classificazione.

Ottimizzazione: L'obiettivo dell'SVM è minimizzare una funzione di costo che include il termine di regolarizzazione (C) e la somma delle funzioni di perdita hinge: $\min_{w,b}$

$\frac{1}{2} ||w||^2 + C \sum_{i=1}^n L(y_i, f(x_i))$ Qui, w sono i pesi associati alle variabili, b è il termine di intercetta, n è il numero di osservazioni e C è un

Module 2: A simple Neural Network (**MLP**)

- Neuron, Layers, Weights

- Loss function

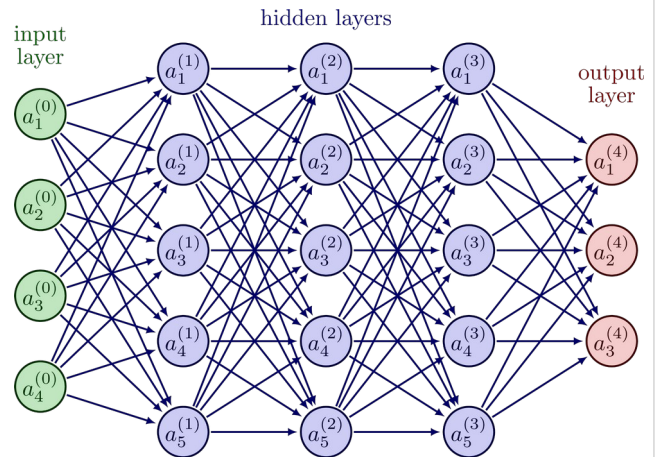
$$L(y, \hat{y})$$

- Feed Forward

$$\hat{y} = f_L(f_{L-1}(\dots f_1(\mathbf{W}_1 \cdot \mathbf{x} + \mathbf{b}_1) \dots) + \mathbf{b}_L)$$

- Backpropagation

$$\frac{\partial L}{\partial \mathbf{W}_i} = \frac{\partial L}{\partial \mathbf{y}} \cdot \frac{\partial \mathbf{y}}{\partial \mathbf{z}_L} \cdot \frac{\partial \mathbf{z}_L}{\partial \mathbf{y}_{L-1}} \cdot \dots \cdot \frac{\partial \mathbf{y}_1}{\partial \mathbf{z}_1} \cdot \frac{\partial \mathbf{z}_1}{\partial \mathbf{W}_i}$$



Le Reti Neurali Artificiali sono modelli di machine learning ispirati al funzionamento del cervello umano. Un'ANN è composta da unità chiamate neuroni, organizzate in strati. Ogni connessione tra neuroni ha un peso associato che viene imparato durante il processo di addestramento. La propagazione all'indietro (backpropagation) è l'algoritmo chiave utilizzato per addestrare reti neurali, e coinvolge il calcolo del gradiente della funzione di perdita rispetto ai pesi.

La funzione di perdita in un problema di classificazione potrebbe essere, ad esempio, l'entropia incrociata o l'errore quadratico medio per un problema di regressione. La denotiamo come $L(y, \hat{y})$, dove y è il valore reale della variabile di risposta e \hat{y} è la predizione della rete neurale.

Dove: \mathbf{x} è l'input della rete, \mathbf{W}_i e \mathbf{b}_i sono i pesi e i bias del i -esimo strato, f_i è la funzione di attivazione del i -esimo strato.

Una volta calcolato il gradiente rispetto ai pesi, i pesi vengono aggiornati utilizzando un algoritmo di ottimizzazione come la discesa del gradiente stocastica (SGD) o varianti più avanzate come Adam.