Machine Learning and AI Presentation

Definizione di Intelligenza Artificiale

Definizione di Machine Learning

Definizione di Deeo Learning

Metodi Statistici

Modelli Predittivi

Funzione polyfit

Funzione mean

Errore Quadratico Medio (MSE - Mean Squared Error)

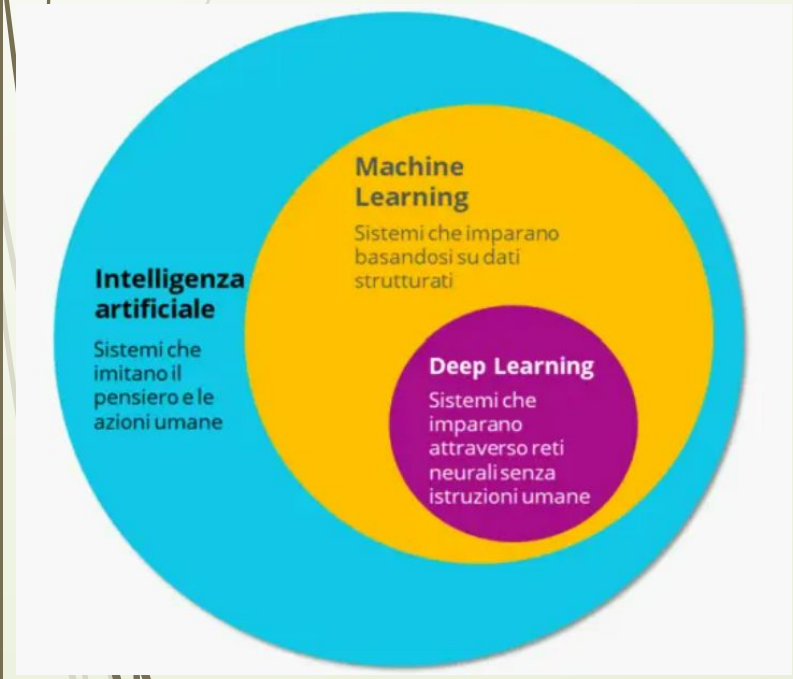
Funzione polinomiale grado n

Codice MATLAB – Progetto fine corso

confronto_modelli_regressione_grafico_csv_do_while.m

Definizione di Intelligenza Artificiale

- L'Intelligenza Artificiale (IA) è il campo dell'informatica che si occupa dello sviluppo di algoritmi e sistemi capaci di simulare comportamenti intelligenti, come il riconoscimento di schemi, il processo decisionale e l'apprendimento, permettendo alle macchine di risolvere problemi complessi o svolgere compiti autonomamente, spesso emulando capacità tipiche dell'intelligenza umana.



Differenze tra Machine Learning e Deep

	Machine Learning	Deep Learning
Formato dati	Dati strutturati	Dati non strutturati
Base di dati	Database controllabile	> 1 milione di punti dati
Addestramento	È necessario un addestratore umano	Sistema di auto-apprendimento
Algoritmo	Algoritmo variabile	Rete neurale di algoritmi
Campo di applicazione	Semplici compiti di routine	Compiti complessi

Definizione di Machine Learning

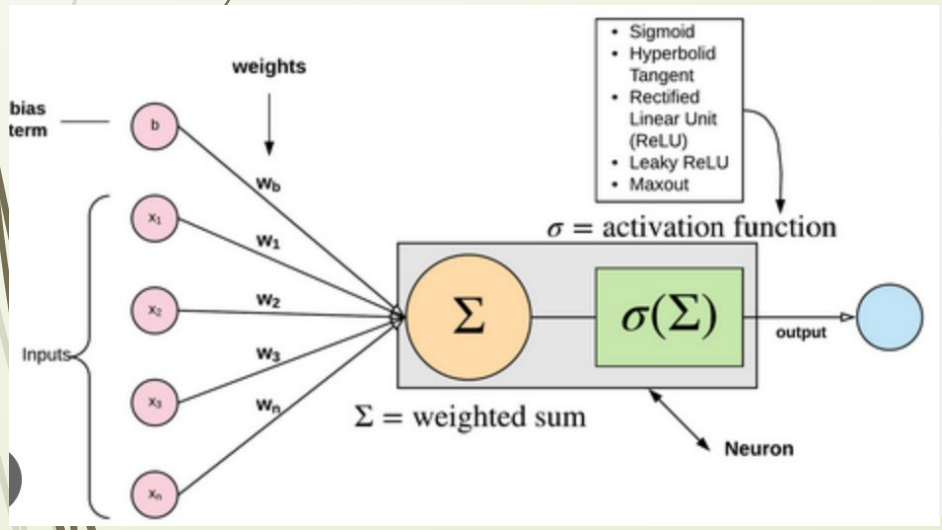
- Il **Machine Learning** (ML) è una branca dell'Intelligenza Artificiale che sviluppa algoritmi capaci di apprendere automaticamente dai dati e migliorare le proprie prestazioni nel tempo senza essere esplicitamente programmati per ogni singola azione. Questi algoritmi analizzano dati, identificano schemi e fanno previsioni o decisioni.
- **Campi di applicazione**:
  - **Riconoscimento vocale e visivo**: usato per assistenti vocali, diagnostica medica, sorveglianza.
  - **Finanza**: previsione di trend di mercato, rilevamento di frodi.
  - **Marketing e pubblicità**: personalizzazione dei contenuti, targeting pubblicitario.
  - **Automazione industriale**: manutenzione predittiva, ottimizzazione della produzione.
  - **Sanità**: diagnosi assistita, scoperta di farmaci.
  - **Veicoli autonomi**: navigazione, prevenzione di collisioni.

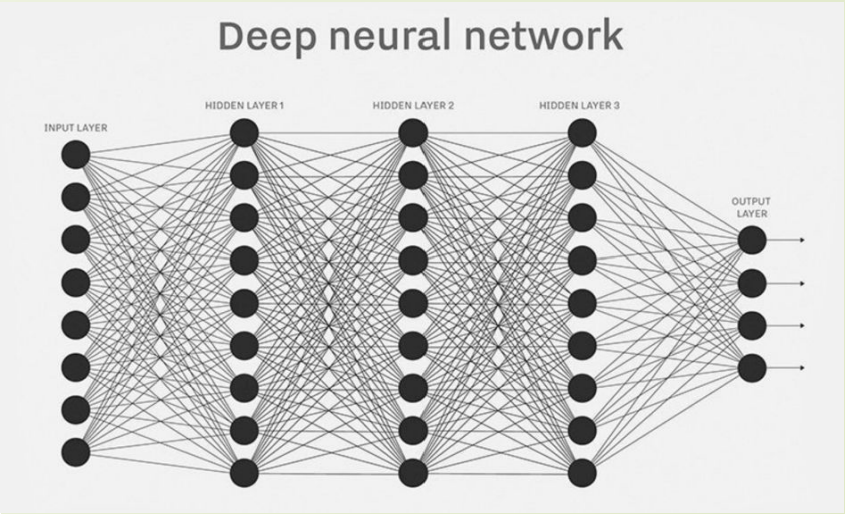
DEEP LEARNING

Il **Deep Learning** è una sottocategoria del Machine Learning che utilizza reti neurali artificiali profonde, cioè con molti strati, per elaborare grandi quantità di dati e imparare modelli complessi. Questo approccio è particolarmente efficace nel riconoscere strutture intricate nei dati, come immagini, testo e audio.

Applicazioni:

- **Visione artificiale**: riconoscimento facciale, diagnostica medica, analisi di immagini.
- **Elaborazione del linguaggio naturale**: traduzione automatica, chatbot.
- **Riconoscimento vocale**: assistenti vocali, trascrizione automatica.
- **Veicoli autonomi**: rilevamento di oggetti, navigazione sicura.
- **Sistemi di raccomandazione**: suggerimenti di prodotti, film, contenuti personalizzati.



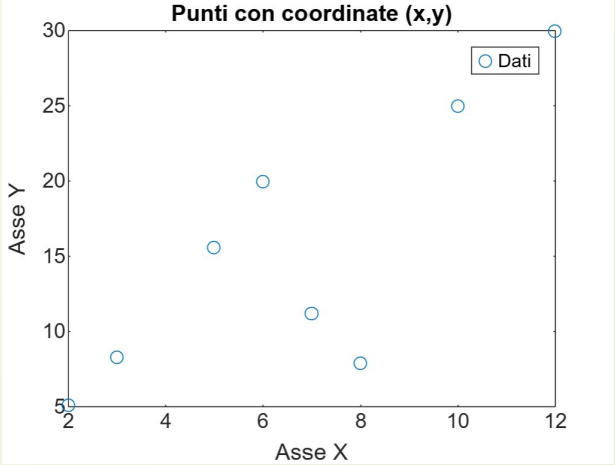


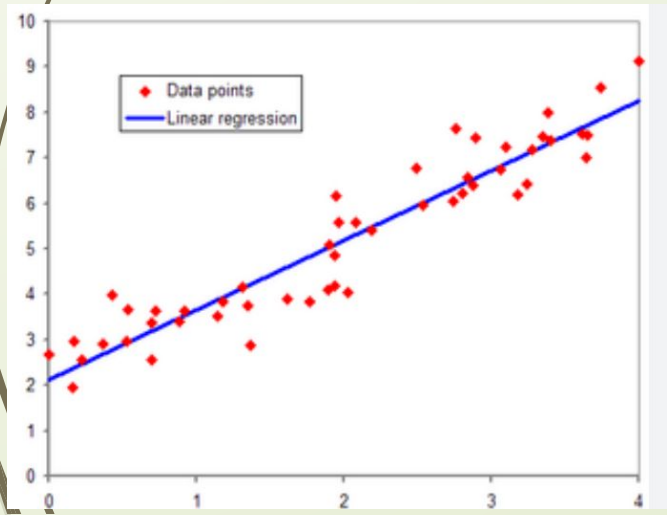
Definizione di Statistica

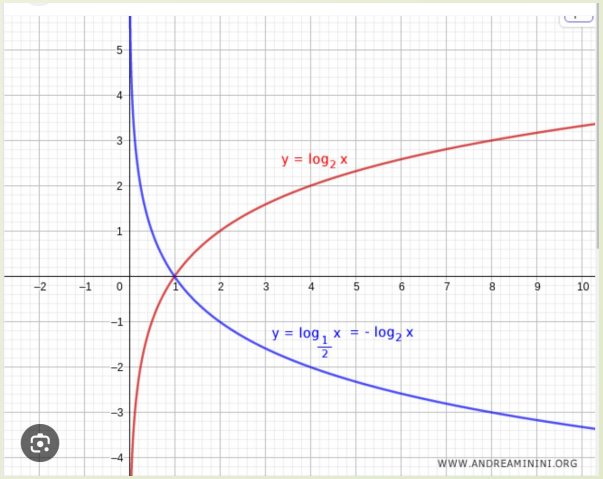
- La statistica è la scienza che si occupa della raccolta, organizzazione, analisi e interpretazione dei dati. In Machine Learning, la statistica è essenziale per comprendere la distribuzione e la variabilità dei dati, nonché per sviluppare modelli predittivi. Le tecniche statistiche come la regressione, l'analisi delle varianze e la correlazione sono alla base di molti algoritmi di ML.

Retta di Regressione e Funzione Logaritmica

- La retta di regressione è un modello statistico che descrive la relazione lineare tra due variabili. È utilizzata per fare previsioni e comprendere l'impatto di una variabile indipendente su una dipendente. La funzione logaritmica, invece, è un modello che cattura relazioni non lineari, come quelle in cui la crescita rallenta nel tempo. Entrambi i modelli sono usati per descrivere dati di diversa natura e ottenere previsioni più accurate.







Che cos'è MATLAB?

- MATLAB è un ambiente di calcolo scientifico e una piattaforma di programmazione ampiamente utilizzata in ambito ingegneristico, scientifico e finanziario. Offre strumenti per l'analisi dei dati, la visualizzazione e la simulazione. MATLAB è spesso usato per implementare algoritmi complessi e modelli matematici, facilitando l'integrazione tra analisi numerica e visualizzazione grafica.

Previsioni usando la Retta di Regressione e Funzione Logaritmica

- La retta di regressione e la funzione logaritmica possono essere utilizzate per effettuare previsioni su variabili continue, come il prezzo di una casa in funzione della superficie. La retta di regressione si adatta a relazioni lineari, mentre la funzione logaritmica è utile per relazioni che presentano tendenze più complesse. In questa fase, si applicano entrambe le funzioni per confrontare i risultati e scegliere la più adatta.

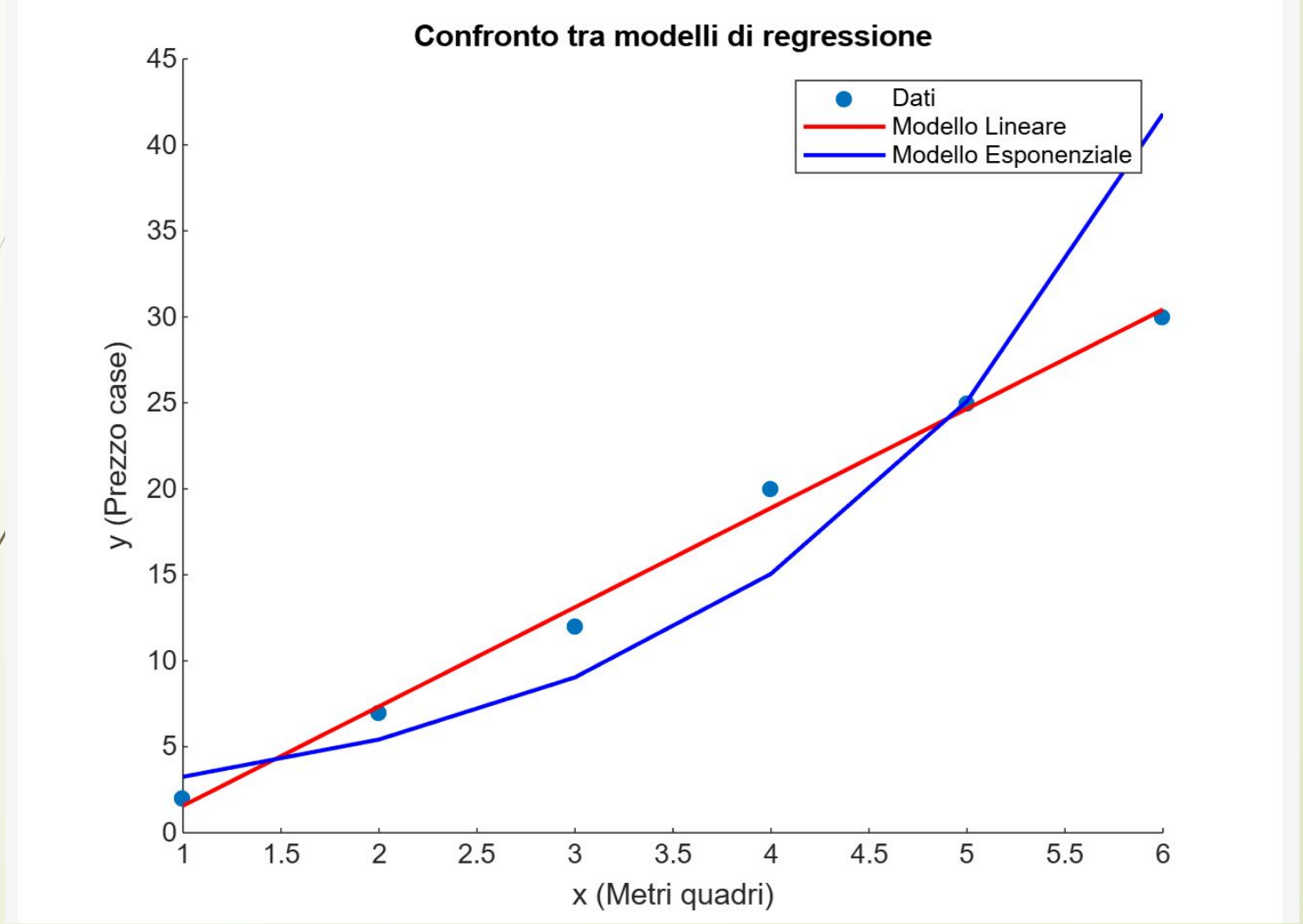
Calcolare quale funzione commette meno errore

- Il confronto tra la retta di regressione e la funzione logaritmica si basa sul calcolo dell'MSE per entrambe. L'algoritmo che produce un MSE più basso è considerato il migliore poiché minimizza la distanza tra le previsioni e i valori effettivi. Questo calcolo è fondamentale per scegliere il modello più accurato per un dato dataset.

Che cos'è un Dataset di Dati?

- Un dataset è una raccolta organizzata di dati strutturati. In Machine Learning, il dataset è utilizzato per addestrare e testare i modelli, permettendo di identificare tendenze e schemi. I dataset possono contenere dati eterogenei, come numeri, testo o immagini, e sono cruciali per l'accuratezza e la generalizzabilità dei modelli.

Dataset:
x1 = [1, 2, 3, 4, 5, 6];
y1 = [2, 7, 12, 20, 25, 30];



Codice MATLAB

- In questa sezione è presentato un esempio di codice MATLAB per la creazione e l'addestramento di modelli di regressione lineare ed esponenziale. Il codice include passaggi per caricare i dati, definire il modello, calcolare le previsioni e misurare l'errore per valutare l'efficacia del modello.
- Esempio di codice MATLAB:

LETTURA DATASET – RETTA DI REGRESSIONE E LOGARITMICA – CALCOLO DELL'ERRORE

```
function confronto_modelli_regressione_grafico_csv_do_while(filename)
    % Funzione per confrontare un modello di regressione lineare ed esponenziale
    % e calcolare il prezzo di una casa dato un valore di metri quadri

    % Lettura del file CSV
    data = readmatrix(filename);

    % Separazione delle colonne (metri quadri e prezzo)
    x = data(:, 1);  % Prima colonna: valori di x (metri quadri)
    y = data(:, 2);  % Seconda colonna: valori di y (prezzo)

    % Modello 1: Regressione lineare
    coeff1 = polyfit(x, y, 1);  % Coefficienti della retta di regressione lineare
    y_fit1 = polyval(coeff1, x);  % Valori stimati con la regressione lineare

    % Modello 2: Regressione esponenziale
    log_y = log(y);  % Trasformazione logaritmica di y
    coeff2 = polyfit(x, log_y, 1);  % Coefficienti della regressione esponenziale
    y_fit2 = exp(polyval(coeff2, x));  % Valori stimati (inverso logaritmo)

    % Calcolo del MSE per entrambi i modelli
    MSE1 = mean((y - y_fit1).^2);  % MSE per il modello lineare
    MSE2 = mean((y - y_fit2).^2);  % MSE per il modello esponenziale

    % Stampa dei risultati MSE
    fprintf('MSE per il modello lineare: %.4f\n', MSE1);
    fprintf('MSE per il modello esponenziale: %.4f\n', MSE2);

    % Confronto tra i due modelli e selezione del migliore
    if MSE1 < MSE2
        fprintf('Il modello migliore è il modello lineare con un MSE di %.4f\n', MSE1);
        modello_migliore = 'lineare';
    else
        fprintf('Il modello migliore è il modello esponenziale con un MSE di %.4f\n', MSE2);
        modello_migliore = 'esponenziale';
    end
```

GRAFICO DELLE FUNZIONI

```
% Plot dei dati e dei due modelli
figure;  % Crea una nuova finestra per il grafico
scatter(x, y, 'filled');  % Grafico a dispersione dei dati originali
hold on;  % Mantieni il grafico aperto per aggiungere altre curve
plot(x, y_fit1, '-r', 'LineWidth', 1.0);  % Aggiungi la retta di regressione lineare (in rosso)
plot(x, y_fit2, '-b', 'LineWidth', 1.0);  % Aggiungi la curva esponenziale (in blu)

% Aggiunta delle etichette agli assi
xlabel('x (Metri quadri)');  % Etichetta per l'asse x
ylabel('y (Prezzo case)');  % Etichetta per l'asse y

% Aggiunta del titolo e della legenda
title('Confronto tra modelli di regressione');  % Titolo del grafico
legend('Dati', 'Modello Lineare', 'Modello Esponenziale');  % Legenda

hold off;  % Rilascia il controllo sul grafico
```

CALCOLO DEL PREZZO STIMATO

```
    % Ciclo do-while per effettuare altre valutazioni del prezzo
    continue_evaluation = 'y';
    while strcmpi(continue_evaluation, 'y')
        % Richiedi all'utente un input per il valore dei metri quadri
        mq_input = input('Inserisci il valore dei metri quadri per calcolare il prezzo stimato: ');

        % Calcolo del prezzo stimato in base al modello migliore
        if strcmp(modello_migliore, 'lineare')
            prezzo_stimato = polyval(coeff1, mq_input);  % Prezzo stimato con il modello lineare
        else
            prezzo_stimato = exp(polyval(coeff2, mq_input));  % Prezzo stimato con il modello esponenziale
        end

        % Stampa del prezzo stimato in euro e del modello migliore utilizzato
        fprintf('Il prezzo stimato per una casa di %.2f metri quadri è: %.2f € (modello %s)\n', ...
                mq_input, prezzo_stimato, modello_migliore);

        % Chiedi all'utente se vuole eseguire un'altra valutazione
        continue_evaluation = input('Vuoi calcolare il prezzo per un altro valore di metri quadri? (y/n): ', 's');
    end
end
```

Utilizzare MATLAB per predire il prezzo delle case

- Usando il codice implementato in MATLAB, si possono effettuare previsioni sul prezzo delle case. Inserendo come input i metri quadri dell'immobile, il modello di regressione addestrato restituirà il prezzo stimato. Questo processo permette di fare previsioni basate su dati storici e analisi statistiche, applicabili in ambito immobiliare.

FINE PRESENTAZIONE

Dimensione applicativa della Matematica: MatLab for STEAM"!

Ringrazio:

Prof.ssa Maria Donatella Fasano
Prof. Francesco Paolo Caforio

Per la professionalità, la competenza e la costante disponibilità dimostrata durante tutto il corso, che hanno contribuito in modo significativo alla mia crescita professionale e personale.

Glossario delle Funzioni: Definizioni e Descrizioni

La funzione **polyfit** è utilizzata per eseguire la regressione polinomiale su un insieme di dati. Questa funzione trova i coefficienti di un polinomio che meglio approssima i dati utilizzando il metodo dei minimi quadrati.

Sintassi di Base:

p = polyfit(x, y, n)

- **x** e **y** sono i vettori dei dati da approssimare.
- **n** è il grado del polinomio.
- **p** è un vettore contenente i coefficienti del polinomio di grado n, ordinati dal termine di grado maggiore a quello di grado zero.

Esempio di UtilizzoSupponiamo di avere dei dati x e y e vogliamo trovare un polinomio di grado 2 che approssimi i dati.

$$P(x) = ax^2 + bx + c$$

```
x = [1 2 3 4 5];
y = [5.1 7.9 8.3 11.2 15.6];
% Trova i coefficienti del polinomio di grado 2. I valori di a, b
p = polyfit(x, y, 2);
% Visualizza i coefficienti
disp('Coefficiente del polinomio:');
disp(p);
% Questo codice restituisce un vettore p con i coefficienti del polinomio di grado 2.
```

Visualizzazione della Retta di Regressione

```
% Crea un intervallo di punti x per il grafico della curva
x_fit = linspace(min(x), max(x), 100);
% Calcola i valori corrispondenti di y con il polinomio ottenuto
y_fit = polyval(p, x_fit);
% Visualizza i dati originali e la curva di regressione
plot(x, y, 'o');        % dati originali
hold on;
plot(x_fit, y_fit, '-'); % curva di regressione
hold off;
xlabel('x');
ylabel('y');
title('Regressione Polinomiale con polyfit');
legend('Dati', 'Curva di regressione');
```

- polyfit(x, y, n) trova i coefficienti del polinomio di grado n;
- polyval(p, x_fit) calcola i valori y della curva di regressione utilizzando i coefficienti p;
- plot visualizza i dati originali e la curva approssimata;

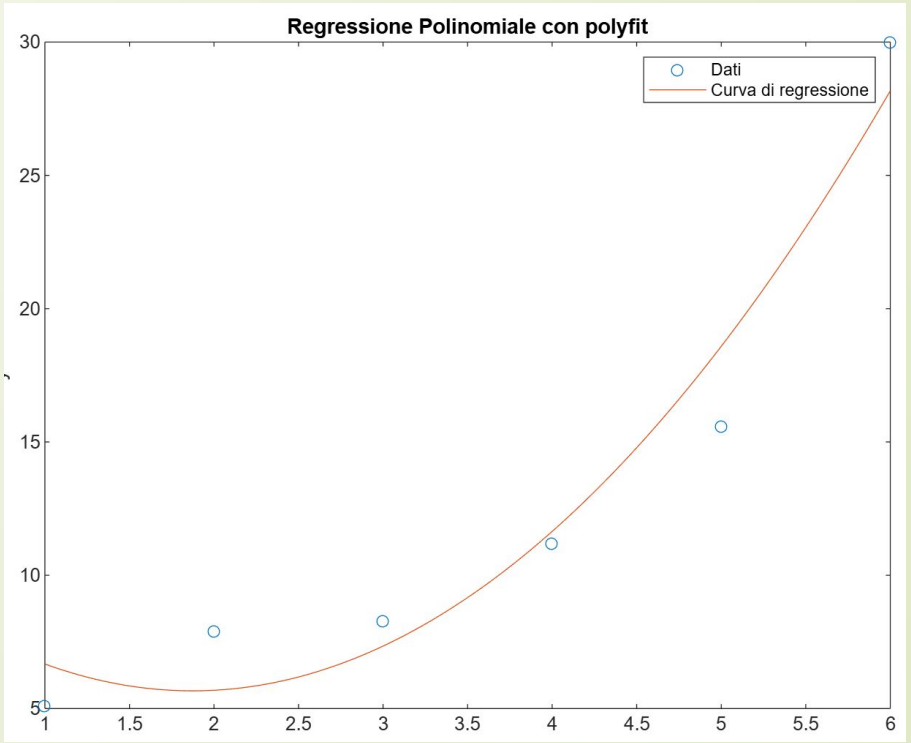
```
x = [1 2 3 4 5 6]; % Corretto l'errore nella definizione di x
y = [5.1 7.9 8.3 11.2 15.6 30];

% Trova i coefficienti del polinomio di grado 2
p = polyfit(x, y, 2);

% Visualizza i coefficienti
disp('Coefficienti del polinomio:');
disp(p);
```

```
>> Polinomio_2_grado
Coefficienti del polinomio:
    1.3214   -4.9500   10.3000
```

Coefficienti del polinomio e rispettivo grafo

/MATLAB Drive/Polinomio_2_grado.m

```
x = [1 2 3 4 5 6]; % Corretto l'errore nella definizione di x
y = [5.1 7.9 8.3 11.2 15.6 30];

% Trova i coefficienti del polinomio di grado 2
p = polyfit(x, y, 2);

% Visualizza i coefficienti
disp('Coefficienti del polinomio:');
disp(p);

% Visualizzazione della Retta di Regressione
% Crea un intervallo di punti x per il grafico della curva
x_fit = linspace(min(x), max(x), 100);
% Calcola i valori corrispondenti di y con il polinomio ottenuto
y_fit = polyval(p, x_fit);

% Visualizza i dati originali e la curva di regressione
figure; % Apre una nuova finestra per il grafico
plot(x, y, 'o');         % dati originali
hold on;
plot(x_fit, y_fit, '-'); % curva di regressione
hold off;

xlabel('x');
ylabel('y');
title('Regressione Polinomiale con polyfit');
legend('Dati', 'Curva di regressione');
```



Il **metodo dei minimi quadrati** è una tecnica matematica usata per trovare la funzione (come una retta o un polinomio) che meglio approssima un insieme di dati. Lo scopo è minimizzare la somma dei quadrati delle differenze tra i valori osservati (i dati) e i valori previsti dalla funzione.

Come funziona

Per un insieme di punti $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, si cerca una funzione $f(x)$ tale che la somma degli errori al quadrato sia minima. Gli errori sono le differenze tra i valori osservati y_i e i valori stimati $f(x_i)$.

La funzione errore è:

$$\text{Errore} = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

Minimizzando questa somma, si ottengono i coefficienti della funzione (ad esempio, i coefficienti di una retta o di un polinomio) che rendono l'approssimazione il più accurata possibile.

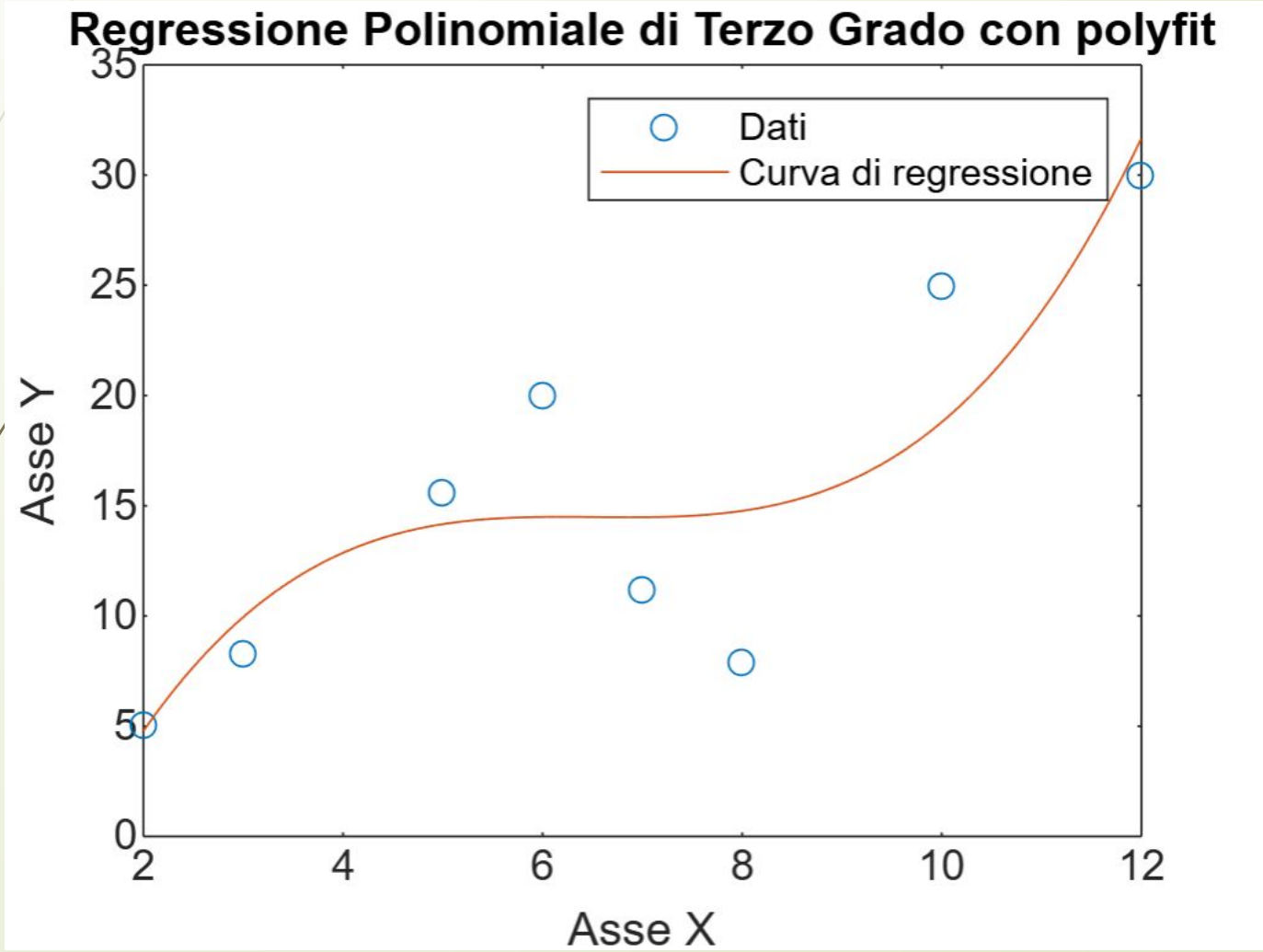
Caso di una retta di regressione (grado 1)

Nel caso di una retta, la funzione $f(x) = ax + b$ ha due coefficienti: la pendenza a e l'intercetta b. Il metodo dei minimi quadrati trova i valori di a e b che minimizzano la somma degli errori al quadrato rispetto ai punti dati.

Polinomi di grado superiore

Quando si usa un polinomio di grado maggiore, come nel caso del tuo codice MATLAB con `polyfit(x, y, 2)`, il metodo dei minimi quadrati determina i coefficienti p_2, p_1, p_0 del polinomio di secondo grado $f(x) = p_2x^2 + p_1x + p_0$ che minimizzano la distanza dai dati.

```
% Trova i coefficienti del polinomio di grado 3
p = polyfit(x, y, 3);
% Calcola i coefficienti di un polinomio di terzo grado per i dati (x, y)
```



```
coeff1 = polyfit(x, y, 1);  % Coefficienti della retta di regressione lineare
y_fit1 = polyval(coeff1, x);  % Valori stimati con la regressione lineare
```

polyval(coeff1, x):
Questa funzione MATLAB è utilizzata per calcolare i valori di un polinomio a partire dai suoi coefficienti e dai punti in cui si desidera valutare il polinomio.

Coeff1:
Questo è un vettore contenente i coefficienti del polinomio, in ordine decrescente di potenza. Ad esempio, per un polinomio di secondo grado della forma ax^2+bx+c**ax 2 +bx+c**, il vettore coeff1 potrebbe essere definito come [**a, b, c**].

x:
Questo è un vettore o un array contenente i punti in cui si vogliono calcolare i valori del polinomio. Può contenere più valori, il che significa che la funzione polyval calcolerà i valori corrispondenti per ciascun elemento di x.

y_fit1:
Questa variabile memorizza il risultato della funzione polyval. Contiene i valori stimati del polinomio per ciascun punto specificato in x. Questi valori rappresentano le previsioni della variabile dipendente (in questo caso, la variabile y) in base al modello di regressione lineare definito dai coefficienti coeff1.

In MATLAB, la funzione **mean** viene utilizzata per calcolare la media (o valore medio) degli elementi di un array. Ecco alcune informazioni su come utilizzare questa funzione:

m = mean(A, dim)

Dove A è una matrice e dim è l'asse lungo il quale calcolare la media:

- Se dim = 1, calcola la media di ogni colonna.
- Se dim = 2, calcola la media di ogni riga.

Calcolo della media di un vettore:

```
A = [1, 2, 3, 4, 5];
m = mean(A); % m sarà 3
```

Calcolo della media di una matrice:

```
B = [1, 2, 3; 4, 5, 6; 7, 8, 9];
m_colonne = mean(B);     % media di ogni colonna: [4, 5, 6]
m_righe = mean(B, 2);    % media di ogni riga: [2; 5; 8]
```

L'Errore Quadratico Medio (MSE - Mean Squared Error) è una delle metriche di errore più utilizzate per valutare la qualità di un modello di regressione. Viene calcolato prendendo la media dei quadrati delle differenze tra i valori previsti e i valori effettivi, misurando quindi quanto le predizioni del modello si discostano dai dati reali.

Formu[illegible]

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

dove:

- n è il numero totale di osservazioni,
- y_i è il valore reale della i-esima osservazione,
- $\hat{y}_i$ è il valore predetto dal modello per la i-esima osservazione,
- $y_i - \hat{y}_i$ rappresenta l'**errore** tra il valore reale e quello predetto,
- $(y_i - \hat{y}_i)^2$ è il **quadrato** di questo errore, utile per evitare che gli errori positivi e negativi si annullino a vicenda.

Come si interpreta l'MSE?
L'MSE misura la **media degli errori al quadrato**. Un valore di MSE più basso indica che le previsioni del modello sono più vicine ai valori reali, segnalando una migliore qualità del modello. Un MSE pari a zero significherebbe che le previsioni coincidono perfettamente con i dati reali (caso ideale e raro nella pratica).

Vantaggi e Svantaggi dell'MSE

- **Vantaggi**:
  - Penalizza maggiormente gli errori più grandi (dato che eleva al quadrato), rendendo sensibili i modelli a valori anomali.
  - È differenziabile, quindi può essere utilizzato per ottimizzare algoritmi di machine learning come la discesa del gradiente.
- **Svantaggi**:
  - L'MSE può essere influenzato dai valori estremi (outliers), in quanto gli errori più grandi hanno un impatto maggiore.
  - Non è interpretabile in termini di unità del dato originale, quindi a volte è preferibile usare l'errore quadratico medio radice (RMSE) che restituisce l'errore nelle stesse unità dei dati.

Applicazione nel Machine Learning

Nell'addestramento di un modello, l'obiettivo è minimizzare l'MSE, portando il modello a fare previsioni il più possibile vicine ai valori reali del dataset di allenamento.

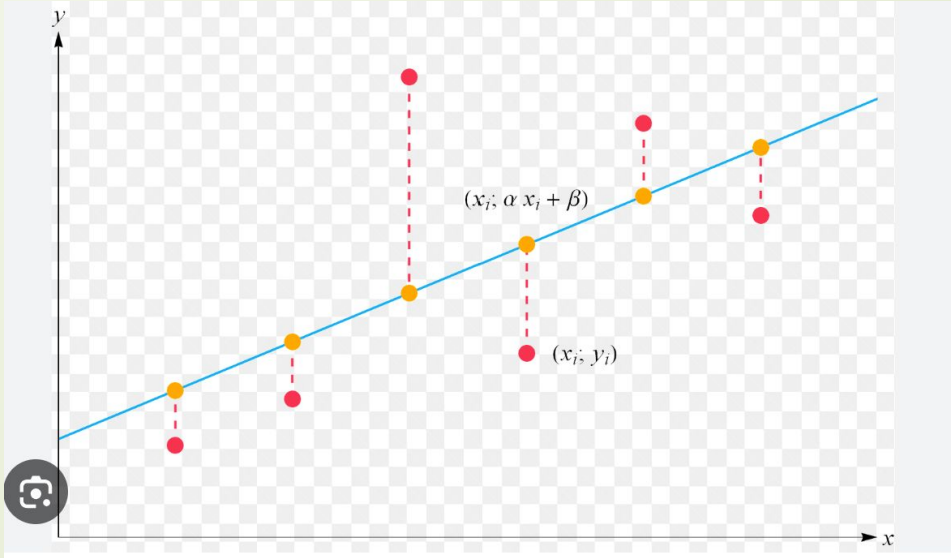
Calcolo dell'Errore - Errore Quadratico Medio (MSE)

- Il Mean Squared Error (MSE) è una misura dell'accuratezza di un modello predittivo. Si calcola prendendo la media dei quadrati delle differenze tra i valori osservati e quelli previsti. Un MSE più basso indica un modello che si avvicina meglio ai valori reali. Questo calcolo è utile per confrontare diversi modelli e determinare quale fornisce previsioni più precise.

- L'Errore Quadratico Medio è una misura comune della qualità di un modello di regressione. È calcolato come segue:

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

  - n è il numero totale di punti dati.
  - yi sono i valori osservati.
  - y^i sono i valori previsti dalla retta di regressione.

- 

```
% Definizione dei dati di input – File Polinomio_3_grado
x = [2 8 3 7 5 6 10 12]; % Coordinata X dei dati
y = [5.1 7.9 8.3 11.2 15.6 20 25 30]; % Coordinata Y dei dati

grado = input('Inserisci il grado del polinomio-->');

% Trova i coefficienti del polinomio di grado 3
p = polyfit(x, y, grado); % Calcola i coefficienti di un polinomio di terzo grado per i dati (x, y)

% Visualizza i coefficienti
disp('Coefficienti del polinomio di terzo grado:');
disp(p);

% Crea un intervallo di punti x per il grafico della curva
x_fit = linspace(min(x), max(x), 100);

% Calcola i valori corrispondenti di y con il polinomio ottenuto
y_fit = polyval(p, x_fit);

% Visualizza i dati originali e la curva di regressione
figure;
plot(x, y, 'o'); % Mostra i dati originali come cerchi ('o')
hold on;
plot(x_fit, y_fit, '-'); % Disegna la curva di regressione come linea continua ('-')
hold off;

xlabel('Asse X');
ylabel('Asse Y');
title('Regressione Polinomiale di Terzo Grado con polyfit');
legend('Dati', 'Curva di regressione');
```

GRUPPO 5

Proff. Caputo, De Luca, Raguso

AREA: *Informatica*

TITOLO

Le funzioni statistiche del Machine-Learning