

# BENCHMARK:

Calcola l'errore di predizione per tutte le settimane dell'Ottobre di un anno specificato (usato come validazione) e ne calcola la media. **Restituisce l'errore percentuale assoluto (MAPE).**

## Contents

---

- [Pulizia dal lavoro precedente](#)
- [IMPOSTAZIONE PARAMETRI](#)
- [Preparazione dei dati](#)
- [STAMPA a schermo degli intervalli di date di identificazione-validazione](#)
- [Stima errore](#)

## Called Functions

---

- [predizione](#)

## Pulizia dal lavoro precedente

---

```
clear all
close all
clc
```

## IMPOSTAZIONE PARAMETRI

---

```
validation_year = 2011;
```

## Preparazione dei dati

---

```
% Caricamento dati di interesse.
load datiOTT
loads = datiOTT(:,2);
years = datiOTT(:,3);

% Estrapolazione delle righe dati dell'ottobre dell'anno di validazione.
% Il benchmark esclude la previsione del 31 ottobre per la sua irregolarita
dati_anno_campione = datiOTT(years == validation_year, :);
numero_giorni_campione = length(dati_anno_campione);
num_settimane_campione = numero_giorni_campione-8; % Escludo ultima settimana
```

## STAMPA a schermo degli intervalli di date di identificazione-validazione

---

```
disp('Anno dei dati di validazione: ')
disp(num2str(validation_year))
% disp('Anni dei dati di identificazione: ')
% disp(strcat(num2str(min(years)), '/', ' ', num2str(max(years(years < validation_year)))))
```

```
Anno dei dati di validazione:
2011
```

## Stima errore

Usando come dati di validazione di volta in volta tutte le possibili settimane di Ottobre dell'anno di validazione, stimo gli errori, salvandoli in un vettore "errore" di cui faccio poi la media.

```
erroreMAPE = zeros(1, num_settimane_campione);
giorno_successivo = zeros(1, num_settimane_campione);
for i = 1:(num_settimane_campione)

    datiWeek = dati_anno_campione(i:(i+6),:);

    L_true = dati_anno_campione(i+7,2);

    L_hat = predizione(datiWeek);

    erroreMAPE(i) = 100*(abs(L_hat - L_true)/L_true);

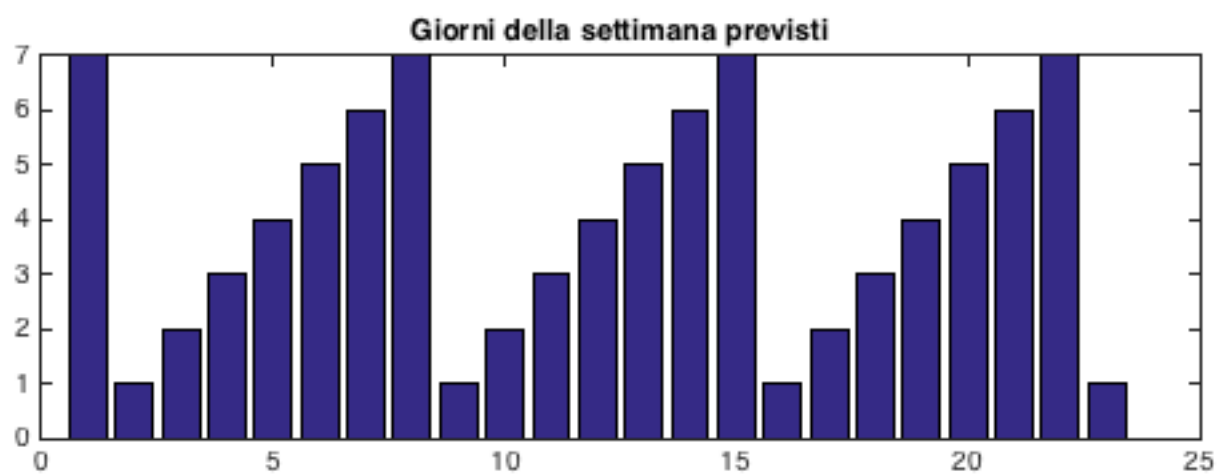
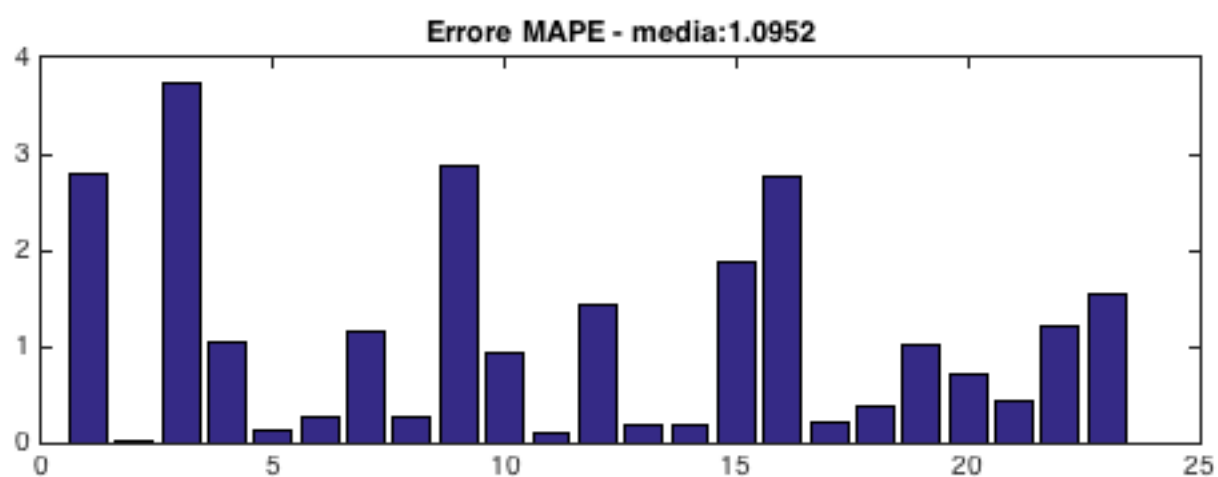
    % Creazione del vettore che immagazzina il giorno per cui è effettuata
    % la previsione
    giorni_settimana = datiWeek(:, 6);
    giorno_successivo(i) = giorni_settimana(7)+1;
    if giorno_successivo(i) > 7,
        giorno_successivo(i) = 1;
    end

end

% Calcolo
mediaMAPE = mean(erroreMAPE);

% Visualizzo il risultato del benchmark in una figura.
figure('NumberTitle', 'off', 'Name', 'Errore MAPE')
subplot(2, 1, 1)
bar(erroreMAPE)
title(strcat('Errore MAPE - media: ', num2str(mediaMAPE)))

subplot(2, 1, 2)
bar(giorno_successivo)
title(strcat('Giorni della settimana previsti'))
```



## Contents

---

- [PREDIZIONE ONE-DAY-AHEAD DEL CARICO ELETTRICO](#)
- [Scelta del numero di anni da escludere dal set di identificazione](#)
- [Import data set e inizializzazioni](#)
- [Elaborazione dei dati degli anni precedenti e stima del modello](#)
- [Elaborazione dei dati della settimana precedente la previsione](#)
- [Predizione](#)

## Called Functions

---

- [destagionalizza](#)
- [detrendizza](#)
- [stima\\_modello](#)

## PREDIZIONE ONE-DAY-AHEAD DEL CARICO ELETTRICO

---

```
function L_hat = predizione(datiWeek)
```

```
%PREDIZIONE Predice il carico elettrico del giorno successivo ad una
%settimana di Ottobre di un dato anno
%   Per la previsione, questa funzione utilizza i dati della settimana di
%   Ottobre precedente al giorno da stimare e un certo numero di mesi di
%   Ottobre di anni precedenti utilizzati come Set di Identificazione del
%   modello previsionale.
%   Dei dati viene effettuata la detrendizzazione, la destagionalizzazione
%   e viene studiato ciò che rimane come modello AR.

% Dati di prova in mancanza di input:
datiWeek = [734799 27930.0408215328 2011 10 23 1;734800 35011.4504166509 2011 10 24 2;73480
1 36994.6270701670 2011 10 25 3;734802 37407.5724480600 2011 10 26 4;734803 36912.883569743
3 2011 10 27 5;734804 36450.5041654192 2011 10 28 6;734805 31192.0431393705 2011 10 29 7];
```

## Scelta del numero di anni da escludere dal set di identificazione

---

```
anni_da_escludere = 1;
```

## Import data set e inizializzazioni

---

```
% Caricamento dati
load datiOTT;
dataSet = datiOTT;

% Restrizione al data set delle righe di identificazione
years = dataSet(:, 3);
boolSet = (years <= max(years)-anni_da_escludere); % Maschera identificazione
dataSet = dataSet(boolSet, :); % Restrizione righe

% Esplicitazione dati di interesse
years = dataSet(:, 3);
```

```
anni_unici = unique(years); % Vettore contenente tutti gli anni diversi
```

## Elaborazione dei dati degli anni precedenti e stima del modello

```
% Detrendizzazione DATI
[loadsDetrended, trend] = detrendizza(dataSet);

% Destagionalizzazione DATI
[loads_deseasonalized, stag_settimanale] = destagionalizza(loadsDetrended, dataSet);

% Stima modello AR sui DATI
[model] = stima_modello(loads_deseasonalized);
```

## Elaborazione dei dati della settimana precedente la previsione

```
giorni_settimana = datiWeek(:, 6);
anno = unique(datiWeek(:, 3));
dati = datiWeek(:, 2); %      log_dati = log(dati);

% Detrendizzazione SETTIMANA
trend_settimana = trend(anni_unici(1: length(trend)) == anno);
if isempty(trend_settimana),
    trend_settimana = mean(dati);
else
    trend_settimana = mean([trend_settimana, mean(dati)]);
end
datiSettimana_detrendizzati = dati - trend_settimana;

% Destagionalizzazione SETTIMANA
dati_destagionalizzati = datiSettimana_detrendizzati; % Vettore che conterra' i dati destag
ionalizzati
for d = 1:7,
    booleanDay = (giorni_settimana == d);
    % Destagionalizzazione
    dati_destagionalizzati = dati_destagionalizzati - stag_settimanale(d)*(booleanDay);
end
```

## Predizione

Predizione basata sul modello estrapolato dai dati degli anni precedenti e calibrato sull'ultima settimana.

```
L_hat_modello = forecast(model, dati_destagionalizzati, 1);
giorno_succ = giorni_settimana(7)+1;
if giorno_succ > 7,
    giorno_succ = 1;
end
L_hat_detrendizzato = L_hat_modello + stag_settimanale(giorno_succ);
L_hat = L_hat_detrendizzato + trend_settimana;
```

```
end
```

```
ans =
```

---

Published with MATLAB® R2015a

```
function [loads_deseasonalized, stag_settimanale] = destagionalizza(loadsDetrended, dataSet)
%DESTAGIONALIZZA Destagionalizza i dati già detrendizzati in ingresso.
%Restituisce i dati destagionalizzati e la stagionalità settimanale
%stimata.
%   La destagionalizzazione dei dati detrendizzati avviene effettuando la
%   media per ogni giorno della settimana di tutti i giorni degli anni
%   che compongono il data set.

dayOfWeek = dataSet(:, 6);

stag_settimanale = zeros(1, 7);
loads_deseasonalized = loadsDetrended; % Vettore che conterra' i dati destagionalizzati
for d = 1:7,
    booleanD = (dayOfWeek == d)'; % Maschera per giorno
    booleanDay = booleanD(1:length(loadsDetrended));
    stag_settimanale(d) = mean(loadsDetrended(booleanDay'));
    % Destagionalizzazione
    loads_deseasonalized = loads_deseasonalized - stag_settimanale(d)*(booleanDay);
end

end
```

---

Published with MATLAB® R2015a

## Contents

- [Estrapolazione dati utili](#)
- [Algoritmo di stima del trend e detrendizzazione](#)

```
function [loadsDetrended, trend] = detrendizza(dataSet)
```

```
%DETRENDIZZA Detrendizza i dati di carico elettrico in ingresso e li
%restituisce unitamente al trend stimato.
%   La detrendizzazione avviene sui dati di carico elettrico contenuti in
%   un matrice nx6, in cui n è il numero di righe (e quindi di dati
%   giornalieri) e 6 sono le colonne contenenti le informazioni sui dati.
%   L'effettivo dato di carico è contenuto nella seconda colonna.
```

## Estrapolazione dati utili

```

loads = dataSet(:,2);
years = dataSet(:, 3);

% Conta degli anni per decidere la lunghezza del ciclo
anni_unici = unique(years); % Vettore contenente un anno diverso per elemento
numero_anni = length(anni_unici); % Numero di anni nei dati

```

## Algoritmo di stima del trend e detrendizzazione

```

trend = zeros(1, numero_anni);
partialMeans = zeros(1, 3); % vettore richiesto per fare la media
loadsDetrended = zeros(1, length(loads));
for i = 1:numero_anni,

    currentYear = min(anni_unici)-1 + i;
    currentYearLoads = loads(years == currentYear);

    % Prendo solo i dati di tutte le sequenze di 28 giorni per l'Ottobre di
    % ogni anno e ne faccio la media (tranne l'ultima, che contiene il 31).
    % In questo modo il trend risulta indipendente da qual'è il primo
    % giorno di ottobre.
    for j = 0:2,
        fourWeeksLoadsPartial = currentYearLoads(1+j:28+j);
        partialMeans(j+1) = mean(fourWeeksLoadsPartial);
    end

    trend(i) = mean(partialMeans);

    % Detrendizzo
    loadsDetrended(years == currentYear) = loads(years == currentYear) - trend(i);

end

```

```

end

```

Published with MATLAB® R2015a

```

function [model] = stima_modello(loads_deseasonalized)
%STIMA_MODELLO Stima di un modello AR sui dati detrendizzati e
%destagionalizzati
% Una volta detrendizzati e destagionalizzati i dati, viene stimato un
% modello AR di settimo grado sui residui. La scelta del grado è stata
% sostanzialmente dettata da una misurazione degli errori e da una
% considerazione pratica: comunque disponendo solo di una settimana
% prima, andare oltre il settimo grado non potrebbe apportare particolari
% benefici

loads_realization = loads_deseasonalized;
model = ar(loads_realization, 7); % AR di settimo grado

end

```