

Consumo di energia: tra statistica e Machine Learning nella previsione di una serie ad alta frequenza

Progetto del corso: *Streaming Data Management and Time Series Analysis*

Rocchi Niccolò 881404

Università degli Studi di Milano Bicocca

Dipartimento di Informatica, Sistemistica e Comunicazione

CdM Data Science F9101Q

23/01/2023

Abstract

La previsione di serie temporali è di fondamentale importanza nel mondo IoT e di monitoraggio delle risorse. In questo studio si confrontano diversi approcci per la modellizzazione e previsione di una serie storica ad alta frequenza riguardante il consumo di energia in Marocco nel 2017. Ognuno degli approcci utilizzati - ARIMA, UCM e Machine Learning - riesce a cogliere diverse particolarità del segnale, ma confrontati nella previsione l'approccio ARIMA è quello che ottiene l'errore medio assoluto più basso.

Keywords: Time Series, Forecast, ARIMA, UCM, Deep Learning, Machine Learning, SVM

1 Introduzione

La modellazione di serie storiche ha trovato ampio spazio negli anni, sia nella ricerca statistica che nel successivo Machine Learning. Questo è dovuto non solo alla possibilità di applicazione in svariati domini - dalla finanza al clima, alla medicina - ma anche per la capacità di riuscire a rispondere a varie domande di ricerca riguardo al mondo che ci circonda. L'extrapolazione di segnale, in senso generico, è infatti sempre più presente nel mondo digitale, e lo studio di nuovi strumenti per lo Streaming Data Management e Analytics diventa fondamentale per dare un senso ai dati. In questo studio si affronta la modellizzazione e successiva previsione della serie temporale riguardante il consumo di energia in Marocco nel 2017. Messi a confronto sullo stesso orizzonte previsivo, si confrontano le diverse capacità di tre approcci: ARIMA, UCM e Machine Learning.

2 Dati e metriche di valutazione

I dati in questione riguardano la serie univariata del consumo di energia nel Marocco dal 01/01/2017 al 30/11/2017. In particolare, la frequenza del dato è

ogni 10 minuti, perciò nello specifico sono presenti 48096 osservazioni, dal 01/01/2017 00:00:00 al 30/11/2017 23:50:00. Lo scopo della ricerca è di prevedere le seguenti 4320 osservazioni, relative all'intero mese di dicembre fino al giorno 30. Chiameremo l'insieme di queste osservazioni test set, poichè sconosciuto.

Prima di iniziare le analisi ci si è accertati che la serie non contenesse valori nulli o mancanti; successivamente è stato isolato come validation set il mese di novembre, tenendo come training set la parte restante della serie. Il migliore modello per ogni tipo di approccio è stato scelto come quello che minimizza il MAE (*mean absolute error*) sul validation set, definito come:

$$\text{MAE}(y, \hat{y}) := \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (1)$$

dove n rappresenta la lunghezza della serie y e della sua previsione \hat{y} .

Successivamente, ogni modello è stato stimato nuovamente sull'intero set di dati a disposizione per ottenere le previsioni del test set. L'obiettivo della ricerca è infatti la previsione e non la modellizzazione ottima del processo generativo dei dati a disposizione.

Un'osservazione riguarda la scelta dell'insieme di dati

usato per calcolare l'ottimalità dei modelli. Tradizionalmente negli approcci statistici si è soliti valutare il modello sull'intero validation set senza passare per una k-folds cross validation. Con i dati a disposizione avrebbe anche poco senso: dovendo prevedere un intero mese di osservazioni, e avendo a disposizione i restanti undici mesi, sarebbe impossibile eseguire una cross validation con dati di valutazione che non si intersecano. Questo perché ogni fold dovrebbe comprendere un mese anch'esso e i dati rimanenti sarebbero troppi pochi per la stima del modello. Tuttavia, non si può neanche permettere un overlap troppo alto tra i vari insiemi; commettere uno di questi errori porterebbe a bias nella valutazione. Si è dunque deciso di portare avanti l'approccio tradizionale, nei modelli statistici quanto in quelli di Machine Learning, per essere coerentemente confrontabili.

3 Modelli ARIMA

3.1 Stazionarietà e autocorrelazione

Raggruppando i dati per giorno e costruendo lo scatterplot - con retta di regressione - tra media e deviazione standard, non si nota una particolare relazione lineare crescente. Inoltre, il valore ottimale restituito dalla funzione di scelta automatica del parametro di trasformazione di Box Cox restituisce 0.293. Proseguendo le analisi, risulta che entrambe le ipotesi nulle dei test di Dickey Fuller, con un lag massimo di 1, e di KPSS vengono rifiutate ma un ulteriore test di KPSS con una differenziazione di 144 (i dati in una giornata) non viene rifiutata. In conclusione si pensa dunque che la serie è integrata e non presenta una non stazionarietà in varianza. I modelli che seguono partono dunque dalla modellizzazione di questa non stazionarietà stagionale e proseguono con cambi iterativi dei parametri (p, d, q) (P, D, Q) $[S]$ basati sui grafici Acf e Pacf. Oltre a questi, la ricerca dei modelli migliori è stata anche guidata anche dal relativo MAE sul validation set: il modello migliore non sarà infatti quello che riesce ad assorbire più memoria lineare sul training set. Questo sottende il fatto che si vuole evitare una situazione di overfitting e al contrario costruire il modello con la più alta generalizzazione.

3.2 Modelli su serie originale

Costruendo i grafici di autocorrelazione sulla serie differenziata di 144 si è notata la presenza di un MA stagionale nel Pacf plot. Il primo modello valutato è stato dunque:

$$[1] \text{ ARIMA } (0, 0, 0) (0, 1, 1)[144]$$

Questo ottiene un validation MAE di 1410.630 ma un Pacf simile al precedente. Aggiungendo un fattore AR

stagionale non si sono ottenuti benefici in termini di errore. Inoltre non sono supportati dal software in uso lag maggiori di 350, quindi non è stato possibile valutare modelli con valori MA o AR maggiori di 1. Guardando l'Acf plot, invece, è ancora presente molta correlazione. Sono stati dunque valutati i seguenti modelli:

$$[2.0] \text{ ARIMA } (0, 1, 1) (0, 1, 1)[144]$$

$$[2.1] \text{ ARIMA } (1, 1, 1) (0, 1, 1)[144]$$

$$[2.2] \text{ ARIMA } (2, 1, 1) (0, 1, 1)[144]$$

Il modello [2.2] è quello che riesce a catturare globalmente più memoria, con valori massimi assoluti nell'Acf e Pacf di circa 0.06. Valori più alti di p migliorano ulteriormente l'Acf nei primi lag. Tuttavia, tutti questi modelli hanno una capacità previsiva più bassa.

Provando a modellare la stagionalità in maniera differente, sono stati valutati ulteriori modelli aggiungendo 8 sinusoidi di periodo 144 come regressori esterni. Sono stati dunque stimati:

$$[3] \text{ ARIMA } (0, 0, 0) (1, 0, 0)[144]$$

$$[4] \text{ ARIMA } (0, 0, 0) (0, 0, 1)[144]$$

$$[5] \text{ ARIMA } (1, 1, 1) (1, 0, 0)[144]$$

$$[6] \text{ ARIMA } (1, 1, 1) (0, 0, 1)[144]$$

Il migliore si è rivelato il [3] con un MAE di 1185.99.

In conclusione, il modello che meglio estrae la memoria lineare è il [2.2], ma con un MAE sul validation set di 4146.588, mentre quello che ottiene l'errore più basso è il [3].

Ulteriori modelli che tentano di spiegare il trend con funzioni spline non si sono rivelati utili nella previsione. Anche il numero di sinusodi si è rivelato ottimale all'interno dei modelli valutati.

3.3 Modelli su serie aggregata

Siccome i miglioramenti più significativi sono stati ottenuti modellando in modo diverso la stagionalità, si è infine provato ad eliminarla costruendo 24 serie temporali, una per ogni ora del giorno. Per fare questo sono stati raggruppati i dati per ora, prendendo la loro media e assegnandola a quella particolare ora del giorno. Gli Acf e Pacf plot di queste serie sono meno complessi dei precedenti, e si è deciso dunque di modellare ognuna delle 24 serie con un modello

$$[7] \text{ ARIMA } (0, 1, 1) (0, 1, 1)[7]$$

Le previsioni di ognuna - sul training e validation - sono state infine concatenate e la serie con un dato ogni 10 minuti è stato ottenuto costruendo il segmento tra ognuno dei due punti della serie oraria risultante, ed estraendone sei punti equispaziati. Il Pacf dei residui risulta ora molto pulito e la stagionalità eliminata, ma l'Acf è ancora abbastanza complesso. Si è però notato che aumentando il valore di p non migliora né il grafico né l'errore sul

validation.

In conclusione, eliminando la stagionalità giornaliera e anzi modellando quella settimanale, si è riusciti ad ottenere un MAE di 996.3704 soppiantando i modelli sulla serie originale. Ulteriori modelli con sinusoidi non hanno portato miglioramenti, come neanche lo stesso modello con differenti aggregazioni orarie (media o trimmed mean) in input.

3.4 Modello finale e conclusioni

Rimanendo sempre nel mondo dei modelli lineari ARIMA, ci si è chiesti se una combinazione lineare - che somma ad 1 - di due modelli differenti avrebbe potuto portare ulteriori riduzioni dell'errore. Così infatti è stato: tramite un ottimizzatore basato sui gradienti coniugati si è trovato il peso w^* tale che:

$$w^* := \underset{w \in [0,1]}{\operatorname{argmin}} \operatorname{MAE}(y, w \cdot \hat{y}_1 + (1 - w) \cdot \hat{y}_2)$$

con \hat{y}_1 e \hat{y}_2 previsioni rispettivamente del modello su serie aggregata e sue serie originale. Questa idea è nata dal fatto che gli errori dei due modelli sono dovuti a cause differenti, e quindi si elidono a vicenda. Si è inoltre notato che i due modelli che soddisfano questa proprietà sono, in ordine: il migliore modello su serie aggregata e il [5]. Questo è dovuto al fatto che il secondo, sottostimando la previsione, riesce ad abbassare le previsioni del primo. Ciò non succede con altri modelli su serie originale, i cui errori sono dovuti a previsioni sovrastimate. Questo si è tradotto in un peso ottimo di 0.771 e un MAE finale di 907.096.

I risultati più significativi dei modelli ARIMA sono riassunti in Figura 3.4. In ordine i modelli: [2,2], [3], [7], [5] + [7]. Il secondo e il terzo sono i migliori modelli rispettivamente sulla serie originale e aggregata, mentre il primo quello che estrae meglio la memoria del training set: è chiaro l'overfitting e la bassa generalizzazione.

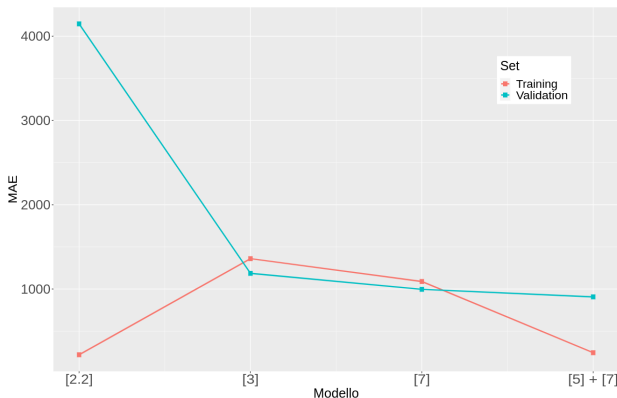


Figura 1: Confronto tra modelli ARIMA

4 Modelli UCM

Osservando la serie storica si nota come le componenti minime di cui ogni modello UCM deve essere composto sono trend e stagionalità. Provando a stimare un modello con almeno queste due componenti, dove la stagionalità può essere espressa con dummy o sinusoidi stocastiche, ci si accorge presto che si necessita di un altissimo tempo computazionale. I modelli tenuti in considerazione sono dunque solo quelli stimati sull'aggregazione della serie. Si è deciso di considerare un'aggregazione oraria calcolata con la media, come in 3.3.

4.1 Modelli su serie aggregata

I primi modelli studiati cercano di modellare la serie oraria come un Local Linear Trend in aggiunta a 24 dummy stocastiche oppure 24 sinusoidi stocastiche. Il modello più coerente è quello a dummy stocastiche, poichè il secondo ottiene stime sfalsate dei picchi di minimo nel validation set, ovvero li prevede in un tempo diverso. Tuttavia, pur essendo coerente con l'andamento generale del validation, il primo modello restituisce previsioni che seguono un trend crescente: questo naturalmente porta ad un MAE elevato, e non si è riusciti a migliorare il comportamento variando i valori dei parametri iniziali passati alla funzione di ottimizzazione.

Si è dunque deciso di partire dal primo modello ed aggiungere un ciclo settimanale stocastico, il cui scopo sarebbe stato quello di eliminare il trend crescente nelle previsioni. Così è stato, ma non solo: il Local Liner Trend del training set appariva meno liscio del precedente, ed essendo una serie persistente questo permette di adattarsi meglio a cambiamenti repentini del trend, traducendosi in una migliore stima del trend finale del training set e dunque una stima migliore di quello del validation. Successivamente si è cercato di adattare i parametri iniziali della funzione di ottimizzazione, basandosi sugli andamenti del validation set che non erano del tutto colti dalla sua previsione. Si ottiene dunque un modello con un MAE di 1226.638. Come sottolineato nei modelli ARIMA, si cerca un modello che minimizzi l'errore su questo insieme di dati e non che colga al meglio i comportamenti del training set. Il fit su quest'ultimo è di 397.395: i modelli UCM, infatti, non colgono la memoria del processo a meno che non appositamente inserita tramite componenti ARIMA nella forma State Space. A questo proposito si è costruito un modello composto da: Local Linear Trend, 24 sinusoidi, un ciclo settimanale e una componente ARIMA (0,1,2) - costruita basandosi sul Acf e Pacf plot dei residui del modello precedente. Il tentativo non ha però dato risultati migliori.

Il migliore modello rimane quindi quello composto da trend, sinusoidi e un ciclo settimanale con un MAE di 1226.638; l'errore di questo è ancora lontano dal precedente modello ARIMA, sottolineando ancora una vol-

ta il fatto che i modelli UCM sono più adatti ad aumentare la conoscenza sulla serie a disposizione ma poco adatti alla previsione in quanto proiettano in avanti l'andamento passato della serie. Si osserva inoltre una grande sensibilità delle stime ai parametri iniziali passati all'ottimizzatore.

5 Modelli di Machine Learning

5.1 Preprocessing

Gli approcci di Deep Learning, e in generale di Machine Learning, spesso non necessitano di una modellizzazione della stagionalità a priori, ma viene eventualmente appresa nel training. Per questo motivo ogni modello, a parte la variazione degli iperparametri, si distingue principalmente per la forma dei dati in input, che può essere:

1. Serie con frequenza originale
2. Aggregazione oraria in media (24 serie)
3. Campionatura di 144 sotto-serie, ognuna per ogni specifico minuto - ad intervalli di 10 - di ogni giornata

Non c'è stata una vera e propria differenza di performance tra le varie forme di preprocessing, ma al contrario esistono modelli che prediligono l'una o l'altra forma. Ciò che però è stato condiviso da tutti i modelli è stata una standardizzazione gaussiana della serie in input.

Si sottolinea infine il fatto che negli approcci di Deep Learning le previsioni sono state fatte per batch, ovvero ogni previsione di lunghezza uguale al batch si basava sulla precedente ed i suoi ritardi, usati come regressori. Per i modelli SVM, invece, si sono costruiti modelli con un orizzonte previsivo uguale ad 1 e si sono ottenute le previsioni sul validation set in modo ricorsivo, come in ARIMA. Questo è stato fatto per avere modelli confrontabili tra i vari approcci ma anche per riuscire a prevedere il test set, il vero obiettivo.

5.2 Modelli Deep Learning: LSTM e GRU

Le reti ricorrenti LSTM (*Long short-term memory*) sono stati i primi modelli ad essere valutati, ognuna con un MSE (*mean squared error*) come funzione di perdita. Si sono costruiti tre modelli, uno per ogni tipo di preprocessing dei dati, i cui regressori erano solamente i ritardi della serie stessa. Per ognuno, si sono variati gli iperparametri come: batch-size e numero di lag, numero di reti LSTM per ogni modello e numero di unità neuronali per ognuna, dropout, funzioni di attivazione e dimensione degli eventuali strati densi finali. Si è notato che, a differenza degli approcci ARIMA e UCM in cui le previsioni dipendono fortemente dagli ultimi dati del training

set, questi approcci risentivano molto del range dei valori dell'intera serie. Le previsioni sul validation set erano dunque spesso sovrastimate, avendo il validation set una media più bassa del training. Inoltre, si notava che i comportamenti delle reti erano molto dipendenti dalla scelta degli iperparametri, soprattutto nelle previsioni a lungo termine: si è dovuto aggiustare questi ad hoc per evitare che le previsioni si appiattissero o al contrario esplodessero.

Il miglior modello LSTM è risultato quello sulla serie originale con un MAE di 2045.793, così composto:

1. Batch size, lag: 288
2. LSTM con 200 unità
3. Dropout: 0.2
4. LSTM con 100 unità
5. Dropout: 0.2
6. Layer denso con 288 unità

Le funzioni di attivazione della rete sono quelle di default e si è prestato attenzione ad attivare quei parametri responsabili del mantenimento temporale del dato. Le altre reti risultano avere un MAE intorno a 2500.

Il secondo approccio ha riguardato invece reti ricorrenti GRU (*Gated Recurrent Units*). Imitando l'architettura LSTM migliore fino a quel momento, si è allenato un modello composto da due reti GRU, ma i risultati ottenuti sono simili alle altre reti. Tentando di migliorare i risultati si è provato ad inserire regressori esterni, come il giorno della settimana e del mese, oppure a processare ulteriormente la serie in input differenziandola di un ordine 144. L'errore del primo approccio è alto anche successivamente al tuning dei parametri, mentre il secondo si pone a livello delle altre reti, mostrando una sensibilità molto forte a questi parametri, specialmente nel Dropout. Non soddisfatti delle Neural Networks, si è tentato un approccio più classico tramite un modello Support Vector Machine.

5.3 Modelli SVM

Costruendo tre modelli, ognuno per una forma diversa di dati in input, si è notato che il migliore era quello allenato sui dati aggregati in media. Con un lag ottimale di 12 e una funzione kernel lineare, questo riesce a raggiungere un MAE di 1758.417, superando i modelli di Deep Learning. Il comportamento delle previsioni, come già accennato, è fortemente influenzato dal range dei dati di training, nonostante la standardizzazione. Un ulteriore approccio in cui si inseriscono sinusoidi o il giorno della settimana come regressori esterni, non hanno portato benefici.

In conclusione, il migliore modello all'interno del mondo Machine Learning è un Support Vector Machine, con un errore di 1758.417.

6 Riassunto delle previsioni

Si riassumono graficamente i comportamenti delle previsioni sul validation set per ogni migliore modello di ogni approccio.

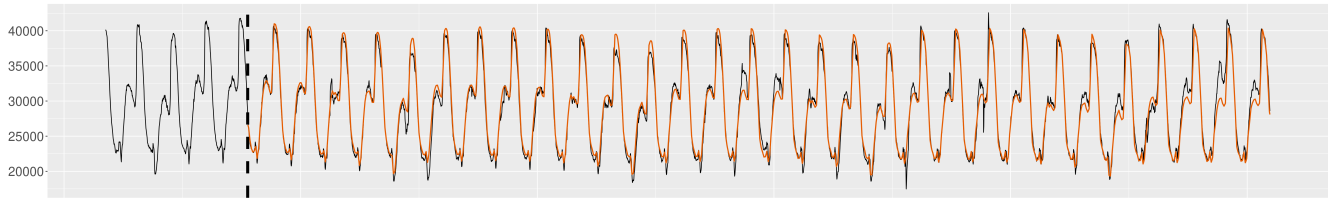


Figura 2: Previsioni del miglior modello ARIMA

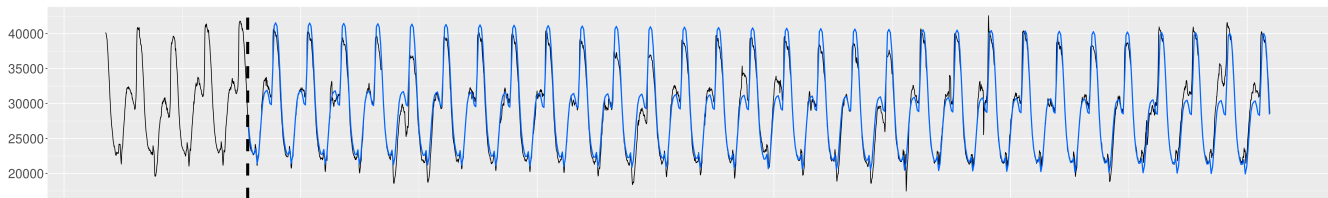


Figura 3: Previsioni del miglior modello UCM

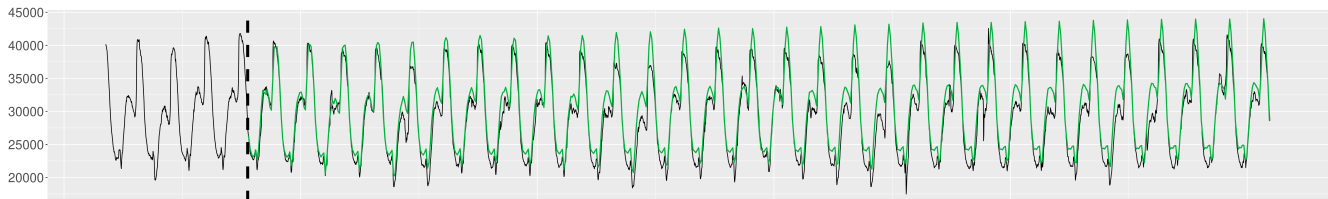


Figura 4: Previsioni del miglior modello di Machine Learning

7 Conclusioni

La serie in esame, come altre serie energetiche o finanziarie, è persistente e ad alta frequenza. Il consumo di energia, d'altronde, dipendendo da molti fattori, è imprevedibile. Questo comportamento la rende di difficile modellazione - e previsione - per tutti i modelli che fanno affidamento sull'intero set di dati e non principalmente sugli ultimi valori, come al contrario succede con ARIMA. I risultati comprensivi ottenuti sul validation set sono espressi in Figura 5.

	MAE
ARIMA	907.096
UCM	1226.638
SVM	1758.417

Figura 5: Risultati finali

Volendo continuare ad indagare e provare nuovi approcci di Machine Learning, è consigliato concentrarsi maggiormente sul preprocessing cercando di rendere la serie più prevedibile. Si dubita che reti più profonde possano risolvere il problema se non si agisce sul preprocessing del loro input, e si pensa anche che si sia ottenuto il massimo dai modelli ARIMA. I modelli UCM, per quanto si posizionino a metà classifica, sono più adatti ad indagare e spiegare la serie che a prevederla.