Università degli Studi di Milano-Bicocca

Streaming Data Management & Time Series Analysis Project

Appliances Energy Prediction

in a Low Energy Building

Autore:
Luca Gabellini, matricola 777786

INTRODUZIONE e DATASET UTILIZZATO

Questo lavoro ha come obiettivo il prevedere i consumi di energia di elettrodomestici in un edificio a basso consumo energetico. Il dataset utilizzato è stato reso disponibile dalla repository UCI[1]. Raccoglie 19375 osservazioni registrate in un arco temporale di circa 4 mesi e mezzo (11 gennaio 2016 – 27 maggio 2016), e 29 variabili.

Per i fini di questo progetto, ovvero l'analisi di una serie storica univariata e il successivo forecasting, sono risultate di interesse solamente due variabili:

- Appliances: consumo di energia di elettrodomestici, registrato in Watt-ora.
- Date: data e ora delle osservazioni, registrate a intervalli di 10 minuti.

Tutte le analisi che verranno presentate sono state condotte con il software R.

PREPROCESSING

Una volta selezionate le due variabili di interesse, i dati sono stati aggregati (sommati) su base oraria. Il dataset risultante è stato poi diviso in training set (primi 3 mesi circa di osservazione, cutoff = 01/04/2016), e test set (rimanente mese e mezzo).

Infine train e test set sono stati trasformati in vere e proprie serie storiche con la funzione xts() di R.

METODI e MODELLI

ARIMA

La prima tipologia di modelli considerata è stata quella degli Arima. Questi modelli richiedono come condizione iniziale la stazionarietà della serie da analizzare. Messa in luce la relazione di linearità tra media (locale) e varianza (locale) della serie, si è provveduto a trasformare quest'ultima su scala logaritimica.

Analizzando volta per volta i correlogrammi, sono stati identificati e stimati in maniera iterativa modelli Arima e Sarima, fino a giungere alla configurazione finale: un Seasonal Arima [2,0,0][1,1,1] con frequenza giornaliera. In figura 1 viene mostrato il grafico della serie (su scala logaritimica) e i valori previsti dal modello sul test.

[1] UCI, Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction#

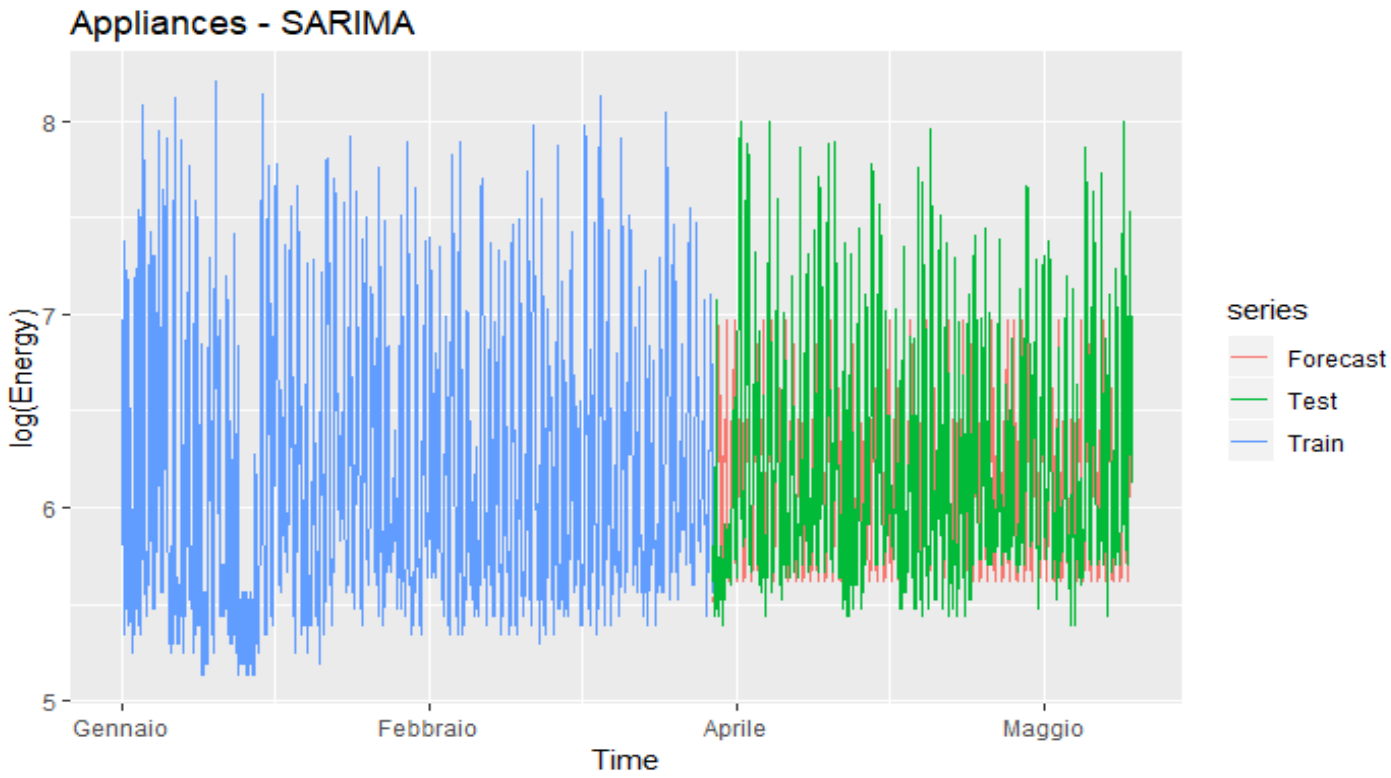

Figura 1 – Sarima Forecasting

Si può notare come il modello preveda discretamente bene i valori del test set ma fatichi a cogliere i numerosi picchi di energia.
I residui del modello, come si evince dalla figura 2, sono approssimativamente normali e quindi white noise.

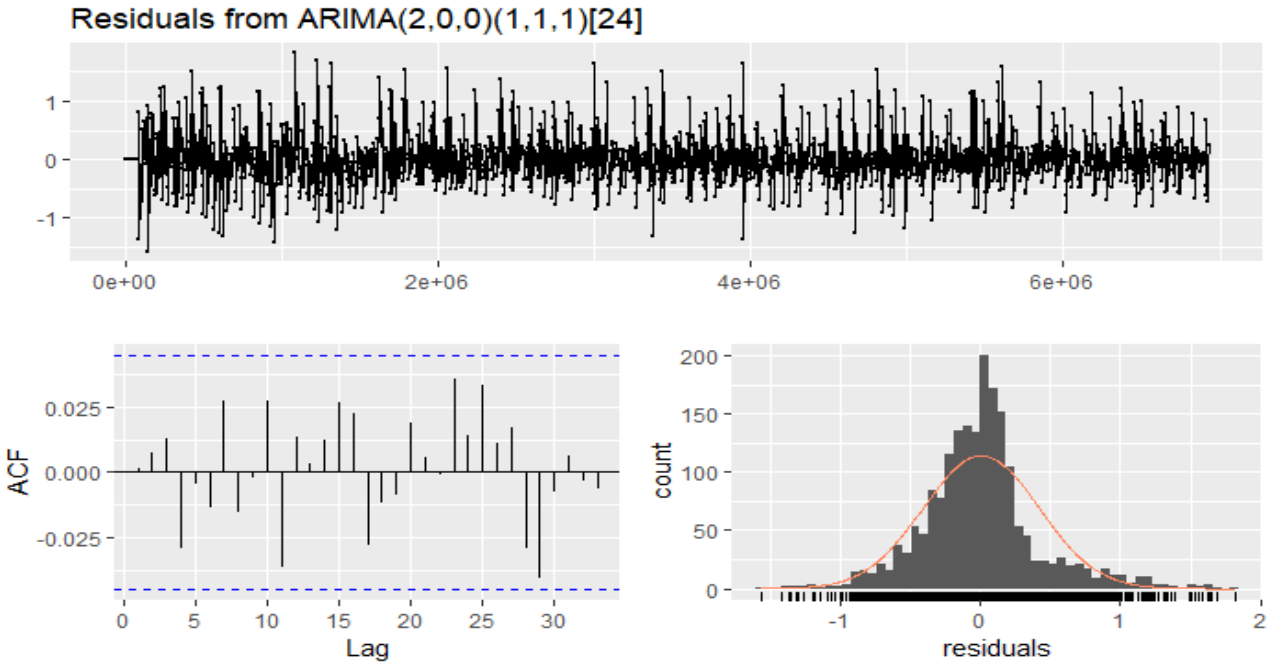

Figura 2 – Arima residuals

UCM

La seconda famiglia di modelli considerata è quella degli UCM, ovvero dei modelli a componenti non osservabili.

Dalle analisi fatte con i modelli Arima e dai grafici è stato visto come la serie rimanga piuttosto costante attorno alla propria media. Di conseguenza si implementerà un modello in forma State Space con componente trend a varianza nulla. Verrà poi inclusa una componente stagionale (giornaliera) a varianza ignota, modellata con 23 dummies.

La serie storica di riferimento per questa analisi e per le successive sarà quella su scala naturale.

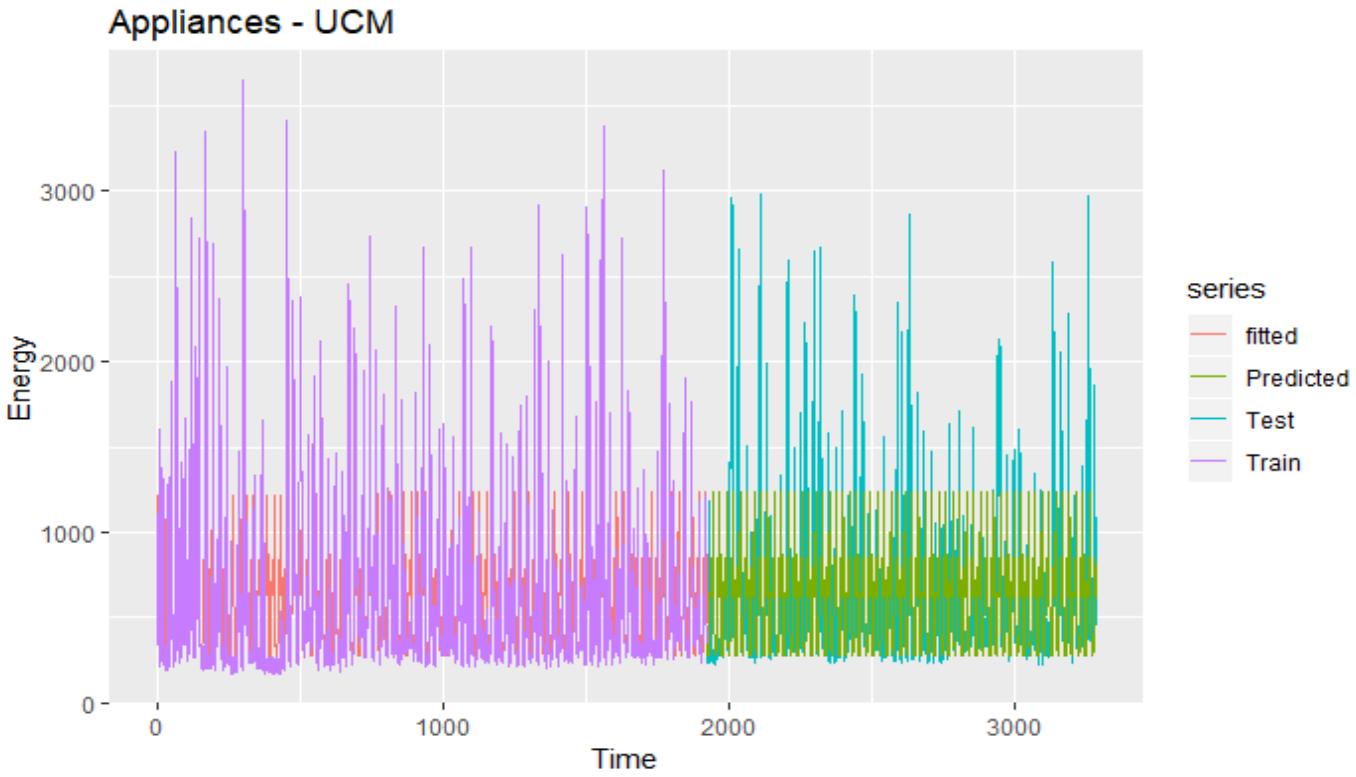

Figura 3: Appliances UCM Forecasting

La previsione sembra quasi riconducibile ad un modello deterministico.

Le previsioni dell'UCM (figura 3) sono simili a quelle del modello ARIMA: il modello fatica a cogliere i numerosi picchi (positivi e negativi) della serie originale.

Per cercare di prevedere meglio i picchi sono state considerate svariate configurazioni di modelli UCM; tra le più significative una con trend costante e due componenti stagionali, una giornaliera, l'altra quadrisettimanale:

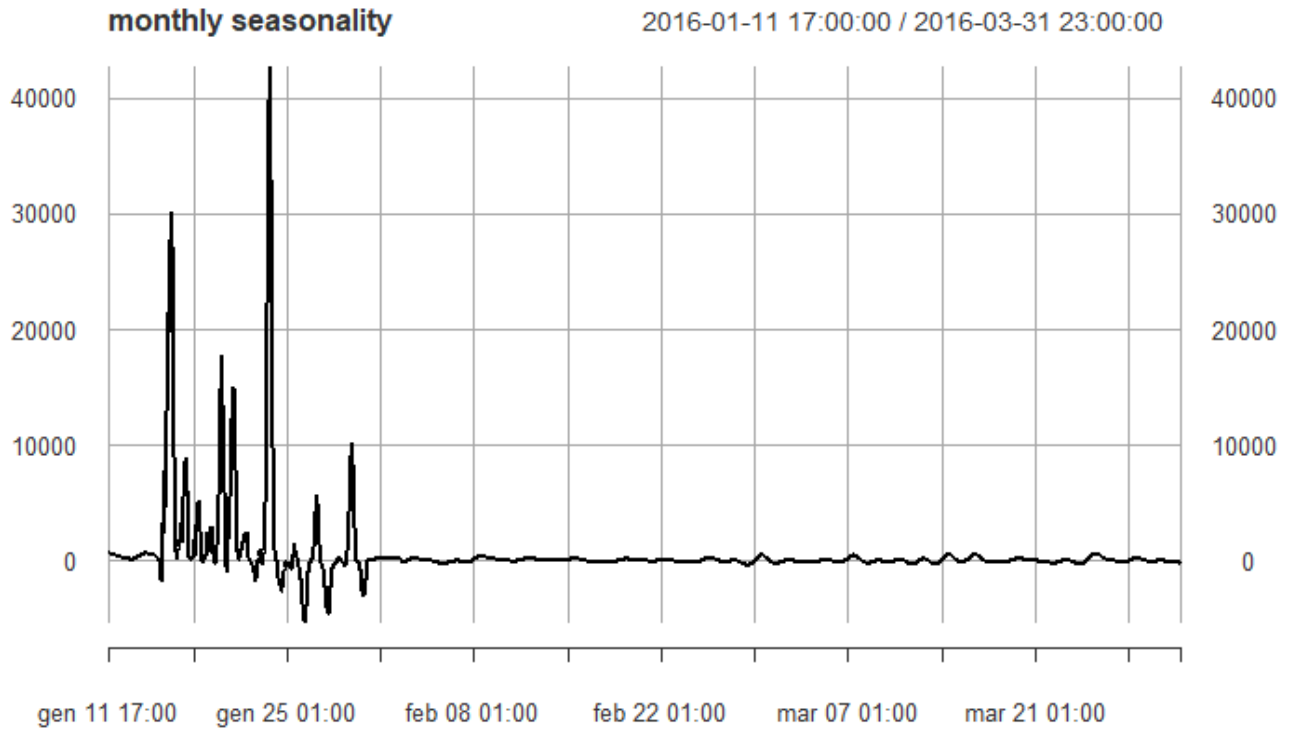

Figura 4 - Stagionalità mensile UCM

La seconda componente stagionale riusciva a cogliere bene i picchi positivi e negativi, in particolare quelli di Gennaio.
La procedura di ottimizzazione del modello andava però a stimare varianze delle componenti enormi, e un modello finale degenere.
Per questo si prende come modello UCM di riferimento il primo, con trend costante e stagionalità giornaliera.

K – NEAREST NEIGHBORS

Per quanto riguarda la parte di machine learning sono stati utilizzati dei modelli KNN.
La funzione knn_forecasting() permette in una sola riga di codice di specificare il modello KNN voluto e fare previsioni.
Il pacchetto è stato reso disponibile agli utenti R di recente, il 31 maggio 2019[2].

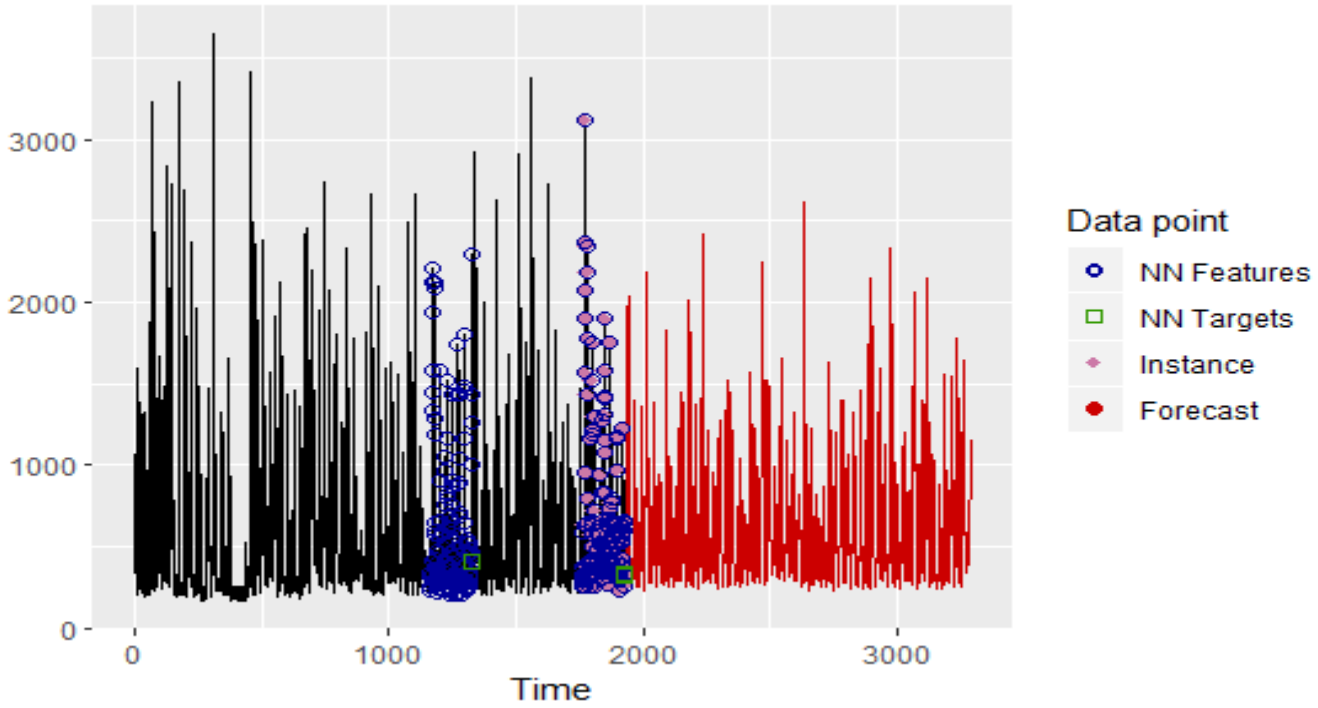

Figura 5 – Procedura di Forecasting KNN

La funzione ha come argomenti principali:

- La serie da prevedere
- Il parametro h, ovvero il numero di step di previsione in avanti
- Lags: numero di lag utilizzati per la previsione
- Il parametro k, ovvero il numero di vicini da considerare
- Msas: la strategia di previsione utilizzata.

Partendo dalla figura 5, viene data una spiegazione pratica sul funzionamento dell'algoritmo:

- Inizialmente viene individuata l'istanza da cui parte la previsione (solitamente le osservazioni finali del dataset) di dimensione equivalente al valore passato al parametro *lags.*

[2] Tsfknn: Time Series Forecasting Using Nearest Neighbors
https://cran.r-project.org/web/packages/tsfknn/

- Successivamente l'algoritmo individua i k vicini più simili all'istanza, visti come vettori di dimensione equivalente all'istanza stessa, in base all'ottimizzazione di una metrica di distanza (la distanza euclidea di default).
- Infine viene fatta una media aritmetica dei target associati ai k vicini, che costituirà il valore della previsione finale.

Il metodo di forecast *'recursive'* permette di calcolare la previsione iniziale un passo in avanti (dimensione del target=1), per poi applicare iterativamente il modello in modo da prevedere tutti i periodi successivi[3].

Sono state provate differenti combinazioni di lags e k (numero di vicini).
Vengono proposti i due modelli migliori:

- knn_m6, con k=3 e lag giornaliero;
- knn_m8, con k=3 e lag settimanale.

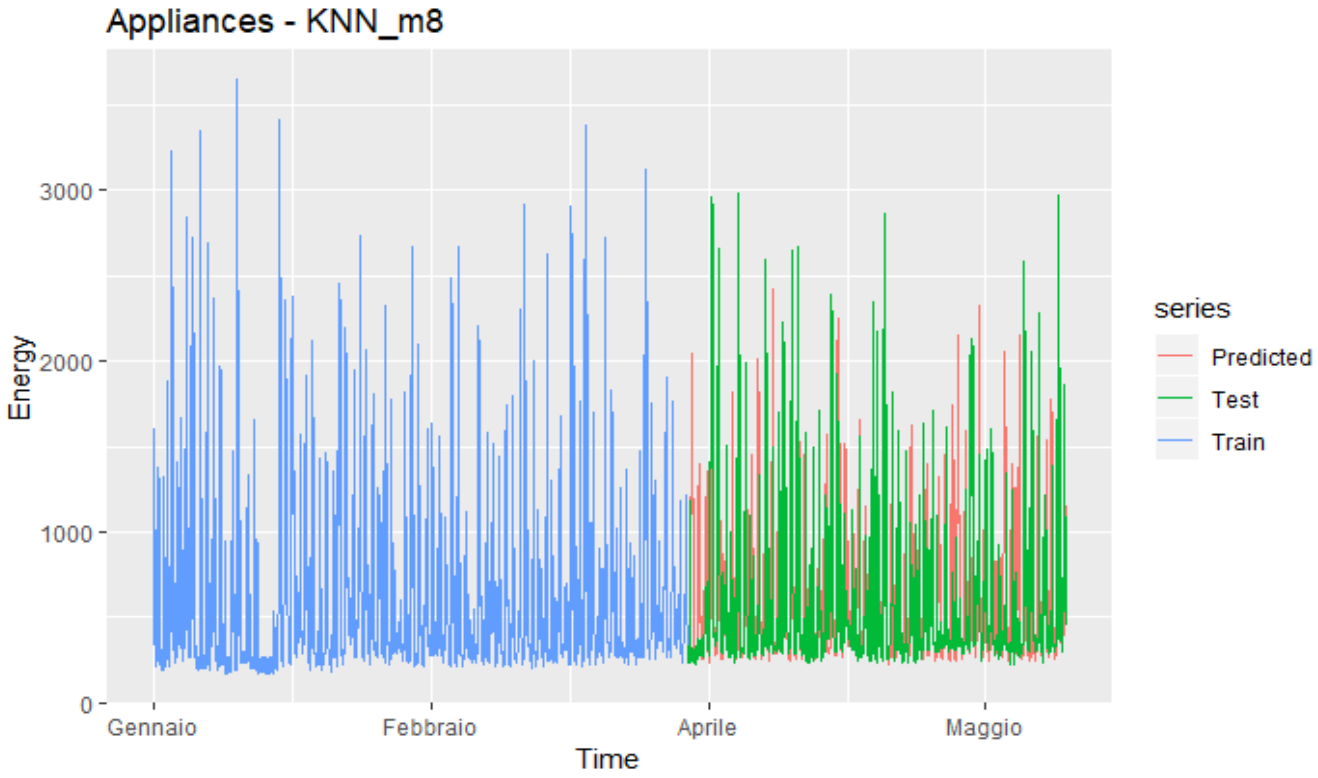

Figura 6 - Forecasting KNN_m8

La figura 6 mostra le previsioni calcolate dall'ultimo modello, che è risultato il migliore tra le varie configurazioni considerate.
Graficamente le previsioni appaiono buone, anche se spesso i picchi previsti non si sovrappongono con quelli reali della serie.

[3] Time Series Forecasting with KNN in R, Francisco Martinez
https://cran.r-project.org/web/packages/tsfknn/vignettes/tsfknn.html

SCELTA DELLA MISURA DI ERRORE

Lo scopo finale di questo progetto consiste nel prevedere i valori di energia elettrica su scala originale. Avendo predetto, con i modelli Arima, valori di energia log-trasformati, si ritiene ragionevole valutare l'errore compiuto in fase di forecasting tra l'esponenziale dei valori predetti e i valori della serie di partenza.

Si assume che la funzione di perdita per i modelli di interesse sia il valore assoluto |e|.
La sopracitata funzione avrà come predittore ottimo la mediana condizionata, invariante per trasformazioni monotone.
Ipotizzando che la distribuzione di interesse (dati log-trasformati) si distribuisca approssimativamente come una normale (fig. 7), media e mediana della distribuzione coincidono, quindi si decide di adottare il Mean Absolute Error come misura di errore.

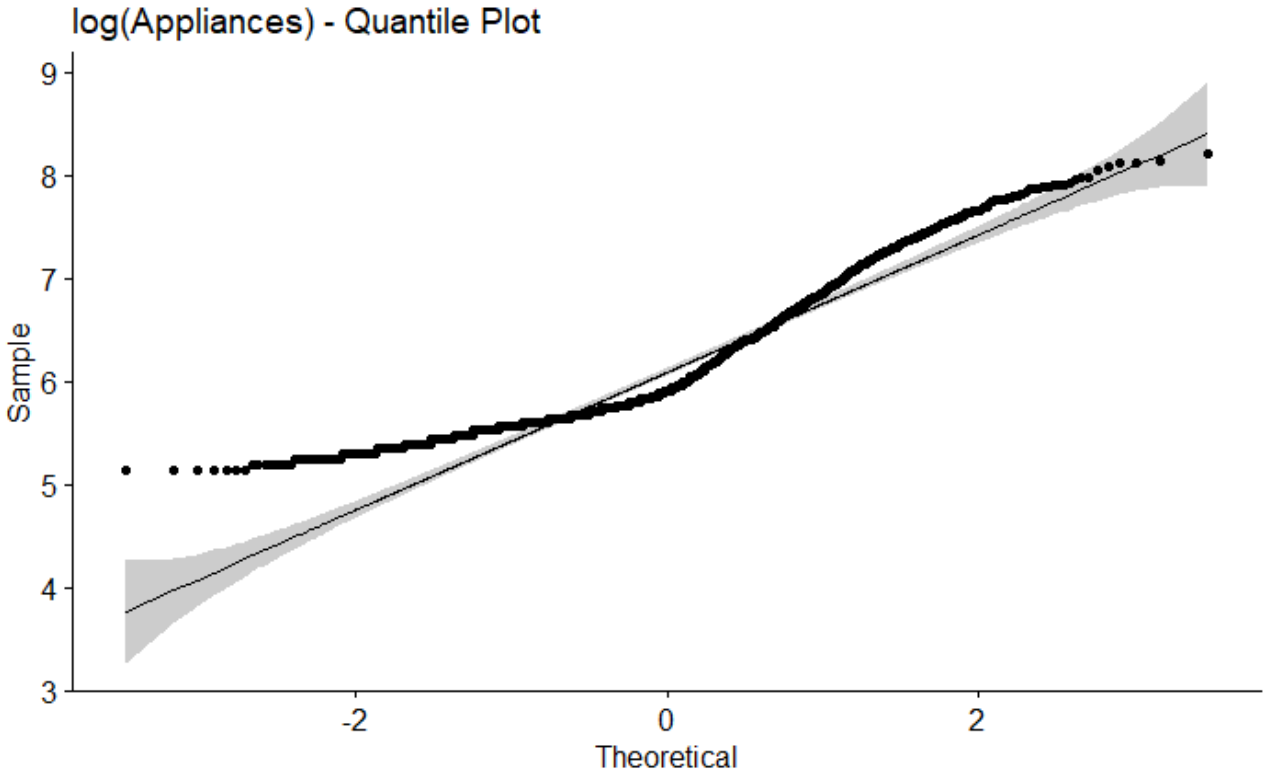

Figura 7 - log(Appliances) QQplot

L'adozione della sopracitata funzione di perdita e annessa misura di errore permette di utilizzare direttamente l'esponenziale delle previsioni senza ricorrere ad alcun tipo di correzione.
Inoltre il MAE risulta direttamente interpretabile, dato che esprime l'errore nella stessa scala in cui sono espressi i dati originali.

CONFRONTO ERRORE TRA STRATEGIE

Nella tabella 1 sono riassunti i valori del Mean Absolute Error dei modelli migliori di ciascuna delle 3 metodologie considerate.

	MAE	
	Train	Test
ARIMA	203.63	260.1
UCM	267.12	296.25
KNN	---	291.63

Tabella 1 - MAE 1

L'Arima sorprendentemente risulta il modello migliore, sia per il fit sul train che per la previsione sul test. A livello grafico il forecasting relativo al KNN lasciava intuire una qualità della previsione migliore rispetto a quella calcolata con il MAE.

Ricordando che si tratta di una metodologia nuova e appena sviluppata, ci si può ritenere soddisfatti dei risultati raggiunti, ancor di più se si confrontano con quelli relativi al modello UCM, che risulta il meno performante tra i tre.

Non è stato possibile calcolare la bontà di fit del modello KNN sul train, dato che l'algoritmo utilizzato non produce valori fitted ma è finalizzato al solo forecasting.

Volendo dare un esempio di interpretazione dei risultati, si può dire che la differenza media assoluta tra i valori di energia previsti in fase di test dal modello ARIMA e quelli reali è di circa 260 Wh.

CONSIDERAZIONI FINALI

La serie storica di interesse è risultata ostica da analizzare, dati i numerosi picchi e pattern complicati da prevedere.

Mi ritengo in ogni caso soddisfatto dei risultati raggiunti, visto e considerato il tempo e le risorse computazionali limitate a disposizione.

A fianco delle tradizionali metodologie per l'analisi e la previsione di serie storiche ne è stata utilizzata una innovativa e poco impiegata in letteratura: ho trovato l'algoritmo KNN interessante e curioso, ideale per questo contesto.

Per analisi successive si propone di impiegare modelli di Machine Learning più sofisticati e "tradizionali" sulla serie, in modo da avere un ulteriore metodo di confronto per le previsioni, che possa giungere a risultati più accurati.