Università degli Studi di Milano-Bicocca

Streaming Data Management & Time Series Analysis Project

Appliance Energy Prediction Dataset

Autore: Matteo Provasi

Matricola 782922

1. Introduzione

L'obiettivo del progetto è quello di predire il consumo energetico di elettrodomestici in un edificio a basso consumo energetico. Il dataset reperibile sulla repository di UCI[1] è composto da 19735 osservazioni e 29 variabili. Oltre alla variabile sul consumo energetico sono presenti delle variabili sulla temperatura e l'umidità nelle stanze dell'appartamento, queste variabili non saranno prese in considerazione. Le ultime variabili sono relative ad un dataset esterno e riguardano le condizioni ambientali, temperatura e velocità del vento, rilevate nell'aeroporto più vicino all'edificio, quello di Chievres[2].

Le osservazioni riguardano il periodo temporale compreso fra 11 gennaio 2016 e 27 maggio 2016, le osservazioni sono rilevate ogni 10 minuti. Per l'analisi dei consumi sono state utilizzate le variabili relative alla data di rilevazione e il valore di consumo energetico.

1.1 Data

Il Belgio come altri paesi europei nell'ultima settimana di marzo passa all'ora legale spostando l'orologio un'ora avanti, passando da CET (Central Europe Time) a CEST (Central Europe Summer Time). Il gap di un'ora dovrebbe verificarsi nella notte fra sabato 26 marzo 2016 e domenica 27 marzo 2016, ma questo non avviene. Nella documentazione del dataset non è presente nessuna informazione a riguardo. Il sito che gestisce le condizioni climatiche dell'aeroporto invece riporta l'orario corretto, però non è possibile verificare quale sia stata la strategia utilizzata dagli autori per la serie storica sui consumi in quanto i dati dalla seconda fonte hanno una frequenza di campionamento diversa da 10 minuti e quindi i dati riportati sono il risultato di un'interpolazione.

Nelle analisi effettuate non è stata apportata nessuna modifica all'orario ma sono stati utilizzati quelli originali presenti nel dataset.

1.2 Pre-processing

Al fine di evitare possibili problemi con la lettura dell'orario da parte di R, l'orario di default è stato fissato su UTC. La variabile contenente la data originariamente era in formato carattere, è stata trasformata in un formato data accettato da R. Per raggruppare le osservazioni a livello orario, sommando di conseguenza i consumi delle sei osservazioni, si è controllato se effettivamente non ci fossero anomalie nella serie. Come detto in precedenza non si verifica nessun gap di orario, mentre l'unica particolarità evidenziata è che l'ultima osservazione è a sé stante, quindi è stata scartata perché impossibile da aggregare con altre.

[1] UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction
[2] Reliable Prognosis, https://rp5.ru/Weather_archive_in_Chievres_(airport),_METAR

1.3 Stazionarietà

Osservando il rapporto fra medie e deviazioni standard della serie si è notato un evidente andamento lineare in cui al crescere della media cresceva anche la deviazione standard. In queste situazioni non è possibile essere certi della stazionarietà in varianza della serie, una delle condizioni necessarie per poter procedere alla creazioni di alcune tipologie di modelli, come quelli ARIMA.

Come trasformata si è scelta quella logaritmo in quanto è noto che per questo tipo di relazioni fra media e deviazione standard il logaritmo è una delle principali soluzioni e anche perché è immediato fare una trasformazione inversa.

Per verificare la stazionarietà in varianza e in media della nuova serie è stato utilizzato il test KPSS che utilizza un modello autoregressivo e i moltiplicatori di Lagrange per la creazione della statistica test. L'ipotesi nulla H_0 è che i dati derivano da un processo stazionario. Per entrambi i test si è ottenuto un valore di $p - value$ maggiore di 0.1 che permette di accettare l'ipotesi nulla.

1.4 Train e test

Il dataset è stato diviso in train e test prendendo come punto di riferimento la data 11 aprile 2016 alle ore 17:00, in quanto la primissima osservazione parte dalle 17:00. Per selezionare l'id del record con l'orario corrispondente è stato applicato un check su R. Nel caso in cui non si specificasse l'orario della sessione all'inizio del codice è possibile che R interpreti la data, senza trasformarla visivamente, nel suo orario corrente. In talune situazioni è quindi possibile che l'id identificato non coincidi con quello reale.

2. Modelli ARIMA

2.1 Train e test

Al fine di identificare i processi che hanno generato la serie da analizzare è stata creata una funzione che mostrasse assieme i grafici relativi all'autocorrelazione (ACF) e all'autocorrelazione parziale (PACF). Il primo identifica il grado di dipendenza fra due valori nella serie storica, il secondo, come suggerisce il nome, considera la relazione tenendo conto dei valori intermedi. Poiché il modello può essere composto da tante diverse componenti, sono stati eseguiti più step con la visualizzazione delle autocorrelazioni per giungere al modello finale. Tutte le stagionalità riportate sono da considerarsi giornaliere; i modelli creati ad ogni step sono stati:

- Step 1: modello $(3,0,0)(0,0,0)$
- Step 2: modello $(3,0,0)(0,1,0)$
- Step 3: modello $(3,0,0)(0,1,1)$
- Step 4: modello $(3,0,0)(1,1,1)$

Sia i modelli 3 e 4 sono stati tenuti in considerazione per un confronto in quanto le autocorrelazioni sono molto simili fra loro. In **Figura 1** sono presentate le analisi sui residui del modello 4.

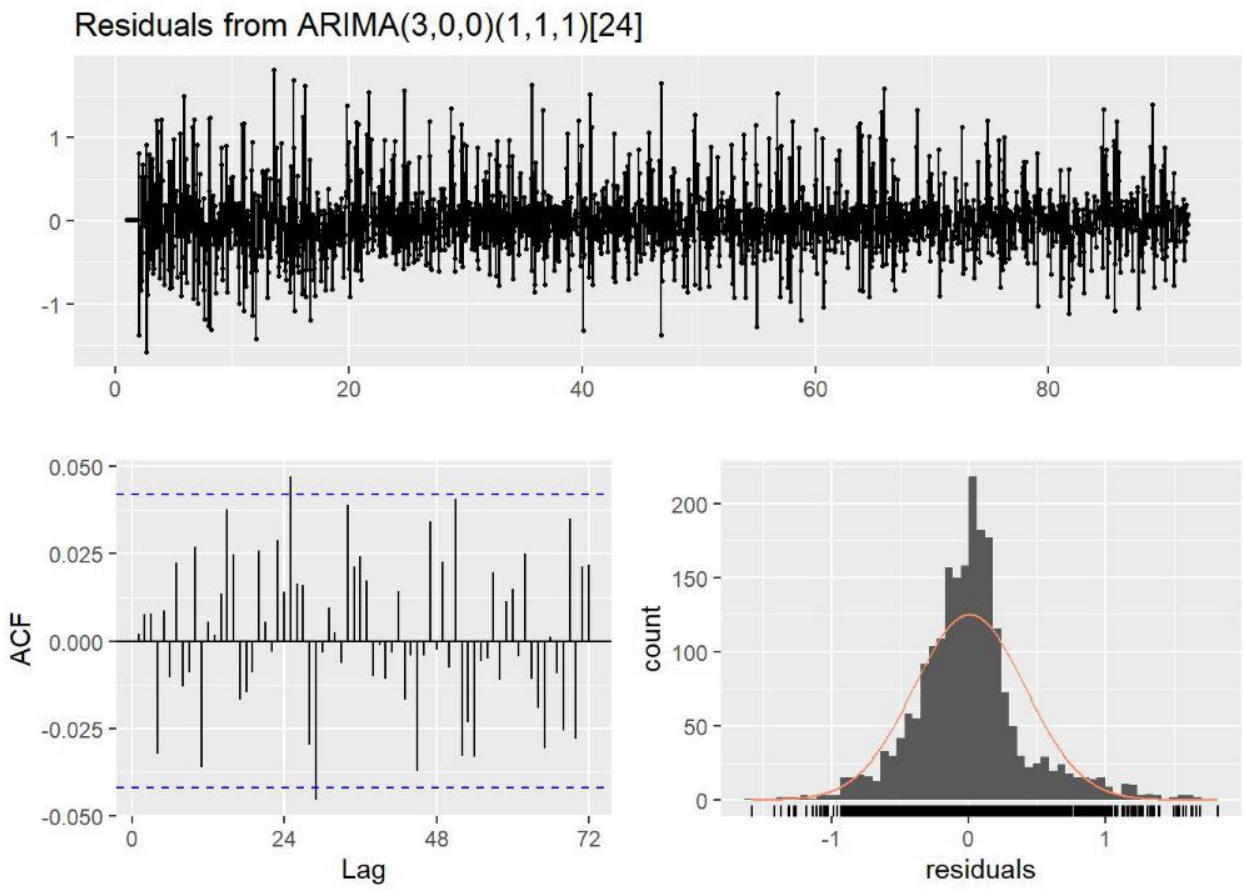

Figura 1: *analisi dei residui del modello ARIMA (3,0,0)(1,1,1)[24]*

Il primo grafico indica i residui delle prime osservazioni, in basso a sinistra si ha l'ACF che era già stato calcolato in precedenza, mentre in basso a destra un istogramma dei residui con disegnata la distribuzione normale. Il test di Ljung-Box serve per comprendere se i residui siano distribuiti in modo indipendente, ipotesi nulla H_0. Il test riporta un $p - value$ ampiamente maggiore di 0.05, che porta ad accettare l'ipotesi nulla.

2.2 Auto-ARIMA

Esiste una funzione in R che permette di trovare il modello ARIMA più appropriato definiti certi parametri. L'algoritmo tuttavia ha restituito un modello che non era buono tanto quanto i precedenti. Una delle possibili motivazioni è che l'algoritmo implementato sia stato nella sua versione approssimata, in quanto i tempi computazionali sarebbero stati estremamente onerosi.

Il criterio per la scelta del modello migliore è l'AIC (Akaike Information Criterion) una statistica che tiene in considerazione il numero di parametri inseriti nel modello e il valore massimo della funzione di verosimiglianza relativa al modello. È una statistica che permette di confrontare diversi modelli

tenendo in considerazione la loro bontà e la parsimonia. In Tabella 1 sono riportati i valori di AIC per tutti i modelli.

Modello	**AIC**
modello $(3,0,0)(0,1,1)$	2368.90
modello $(3,0,0)(0,1,1)$	2362.89
modello $(3,0,1)(2,0,0)$ auto-ARIMA	2579.64

Tabella 1: *confronto fra modelli in base all'AIC*

Più il valore di AIC è basso migliori sono le performance del modello. Il modello ottenuto da auto-ARIMA è il peggiore mentre l'ultimo modello creato al punto precedente risulta essere di poco il migliore.

2.3 Previsioni ARIMA

In Figura 2 è mostrato il plot della serie suddivisa fra train test e previsioni.

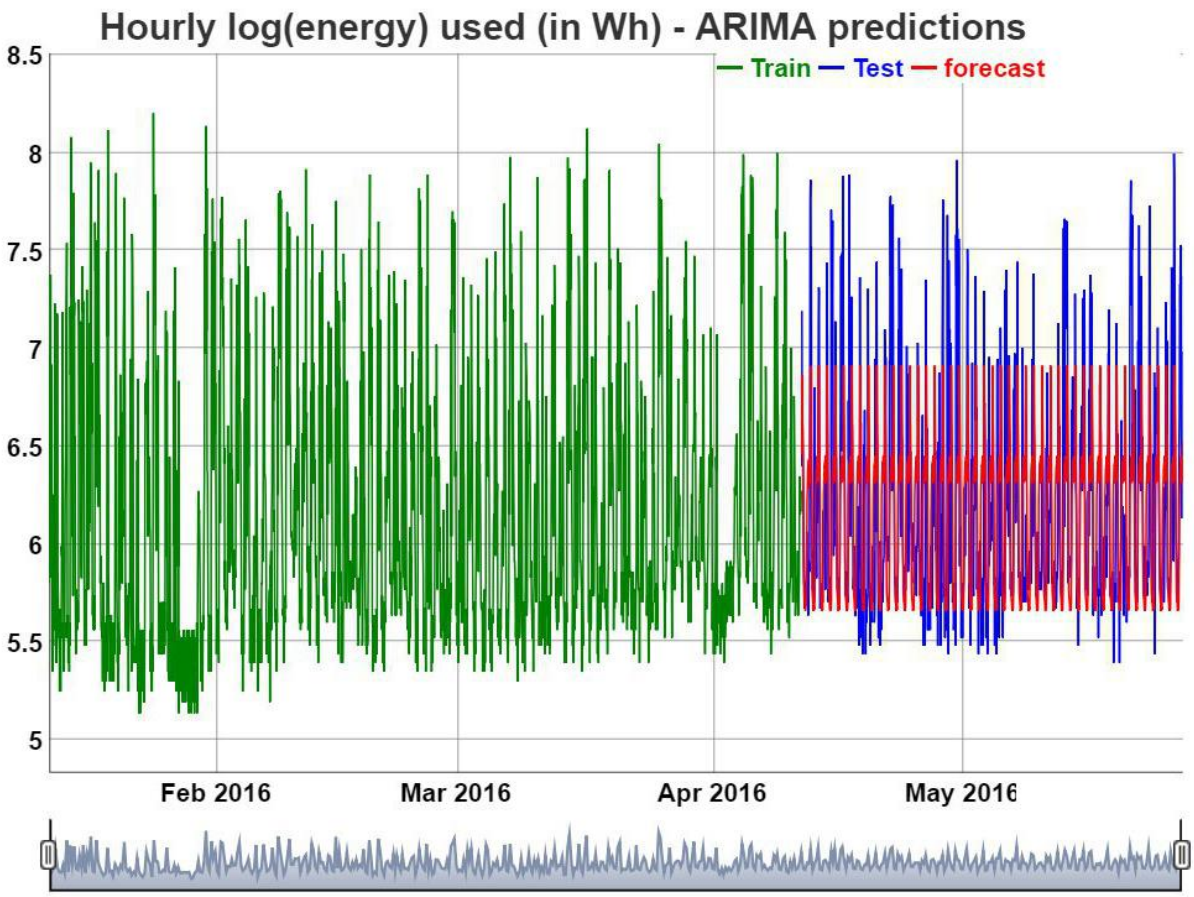

Figura 2: *previsioni del modello ARIMA*

Come si può notare dal grafico, nonostante sia il modello migliore, l'andamento è prettamente deterministico e i picchi, sia in positivo che in negativo non sono colti.

3. Modelli UCM

La seconda categoria di modelli considerati sono quelli UCM, ovvero i modelli a componenti non osservabili. Questi modelli sono creati in R utilizzando la forma State Space e il pacchetto $KFAS$.

3.1 UCM a singola stagionalità

Dalle analisi precedenti è risultato evidente che la media della serie non segua un attrattore ma sembra rimanere costante, di conseguenza il modello implementato avrà un local linear trend con varianza nulla. La seconda componente sarà una stagionalità formata da 23 dummy con varianza ignota che dovrà essere stimata dall'algoritmo. Il modello appena creato aveva un comportamento simile a quello dell'ARIMA precedente con andamento deterministico delle dummy.

3.2 UCM a doppia stagionalità

Per migliorare il modello si è notata una particolarità nella serie logaritmica come rappresentato di seguito in Figura 3.

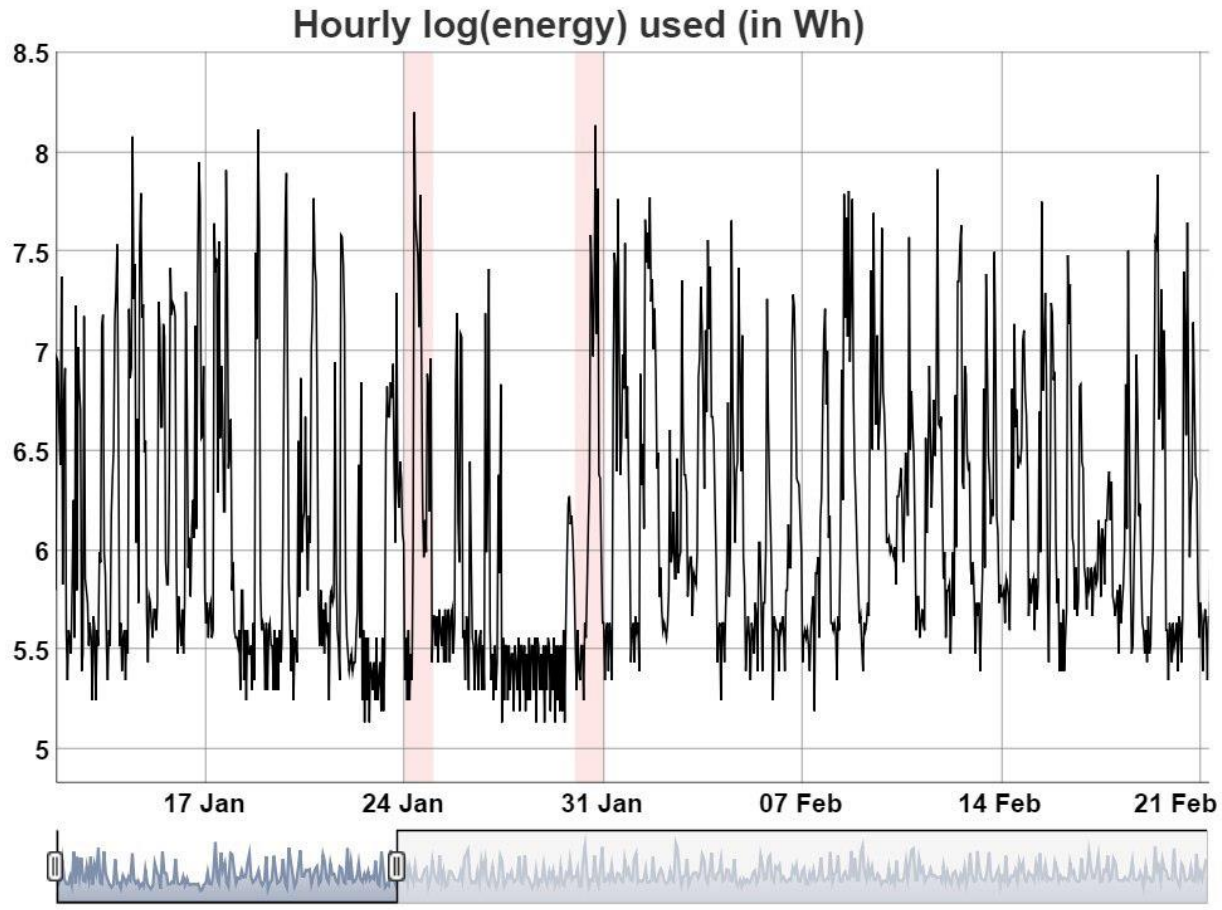

Figura 3: *picchi di energia che si verificano ad una settimana di distanza.*

I principali picchi di energia sul train avvengono ad una settimana di distanza, da questo punto si è partiti con la creazione del secondo modello UCM che ha anche una stagionalità settimanale con sinusoidi. In Figura 4 e Figura 5 sono rappresentati rispettivamente la componente stagionale settimanale e le previsioni del secondo modello UCM.

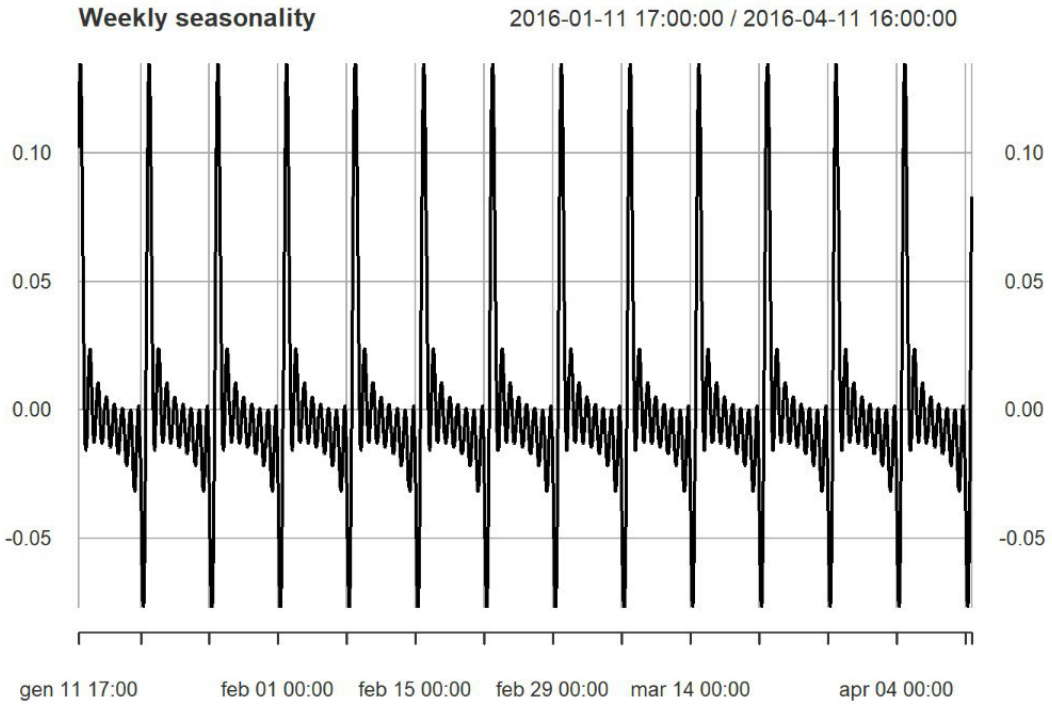

Figura 4: *componente stagionale creata dal modello.*

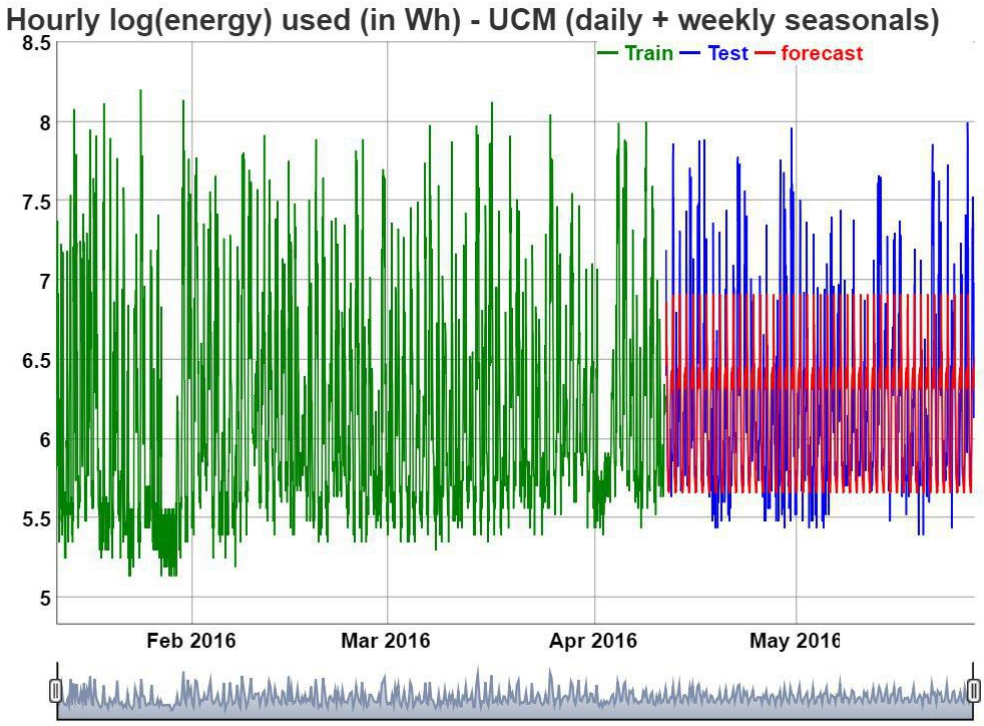

Figura 5: *previsioni del modello UCM*

La componente stagionale settimanale ha un andamento in cui si raggiungono picchi elevati sia in negativo che in positivo. La scala di variazione però è molto contenuta e questo si rispecchia anche nel modello completo che sembra avere ancora la struttura di un modello deterministico. Andando ad osservare nel dettaglio le previsioni si noterà che i picchi più elevati non sono predetti ma per il resto della serie la previsioni sono molto simili alla curva originale.

Uno dei criteri per confrontare le performance di diverse tipologie di modelli è l'utilizzo dell'MAE (Mean Absolute Error):

$$MAE = \frac{\sum_i^n |e_i|}{n}$$

Si è deciso di utilizzare questa statistica in quanto facilmente interpretabile e poiché la trasformazione dell'errore in valori non logaritmici è immediata senza dover applicare delle correzioni.

Dal confronto fra ARIMA e il secondo modello UCM, quest'ultimo è risultato il migliore.

4. Modelli Machine Learning

Per la parte di machine learning sono stati implementati degli algoritmi di kNN (k Nearest Neighbour). L'algoritmo chiede in input il numero di osservazioni che devono essere predette, il lag da considerare e il numero di vicini. L'algoritmo procede nel seguente modo:

- Vengono identificate le istanze che, solitamente, corrispondono alle osservazioni finali del train in quanto la previsione deve "iniziare" da quelle.
- Tramite una funzione di distanza sono identificati i valori più vicini alle istanze.
- I valori prossimi a queste istanze sono chiamati target e dalla media di questi viene effettuata la previsione.

4.1 kNN: scelta di k

La tecnica utilizzata, MIMO (Multiple Inputs Multiple Outputs), richiede che i punti che formano un target siano consecutivi fra di loro e di lunghezza pari a lag specificato. Secondo gli autori il numero consigliato di vicini da utilizzare è, date n osservazioni nel train, pari a $k = \sqrt{n}$.

Il modello con questo numero di vicini sembrava essere molto contenuto per quanto riguarda i valori con il problema comune di mancare l'identificazione dei picchi. Si sono provate quindi differenti combinazioni di parametri e si è notato che al diminuire del numero dei vicini la serie iniziava ad oscillare in maniera molto più evidente. Due dei modelli creati con questo approccio, rispettivamente con 3 e 5 vicini, sono rappresentati in Figura 6.

Facendo un raffronto del MAE dei modelli kNN creati si è riscontrato che in realtà i modelli con pochi vicini erano quelli che performavano peggio.

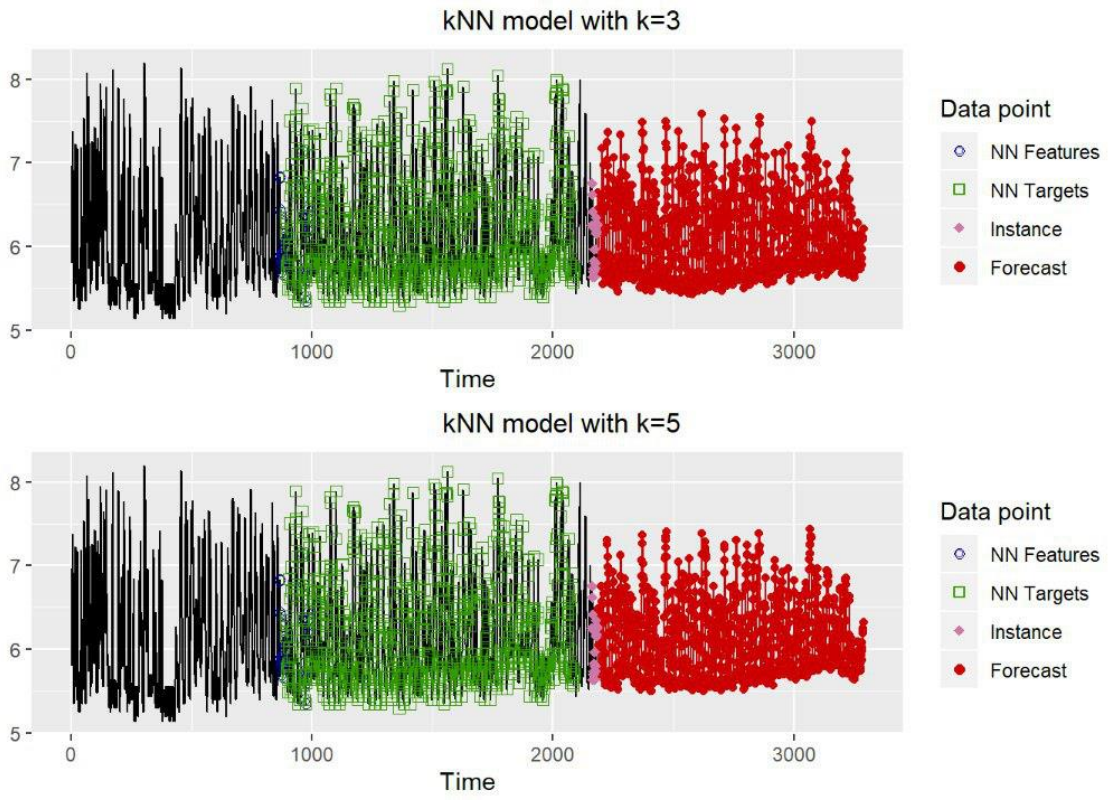

Figura 6: *previsioni del modello UCM*

Come si evidenzia dal plot in Figura 7 il MAE cala quasi linearmente all'aumentare del numero di vicini fino a raggiungere il valore minimo con 41.

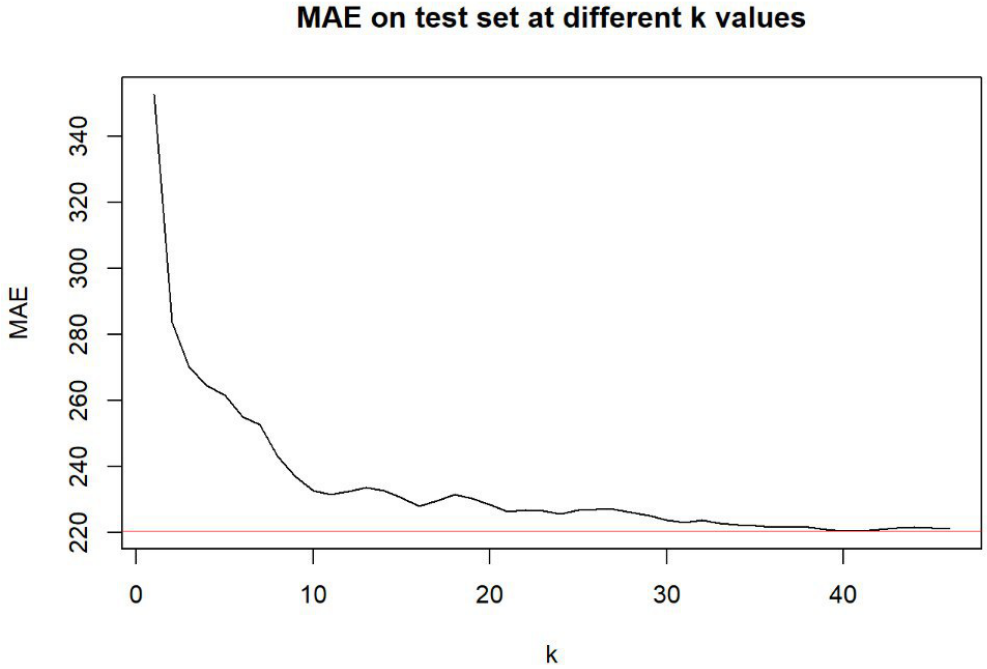

Figura 7: valori del MAE al variare del numero di vicini

Una delle possibili spiegazioni è che i modelli con pochi vicini hanno dei picchi elevati poiché nei target considerati ci sono i picchi che sono mediati con meno osservazioni o con osservazioni riguardanti un alto consumo. Possono quindi esserci delle previsioni con valori elevati ma il corpo centrale della serie, che comprende il maggior numero di osservazioni, non è predetto correttamente e questo porta ad un aumento del MAE.

4.2 kNN con K=41

Il modello con tanti vicini, come rappresentato in Figura 8 ha una struttura che vista complessivamente sembra più rigida, ma nel dettaglio si vede che riesce ad adattarsi molto bene alla serie storica, picchi esclusi.

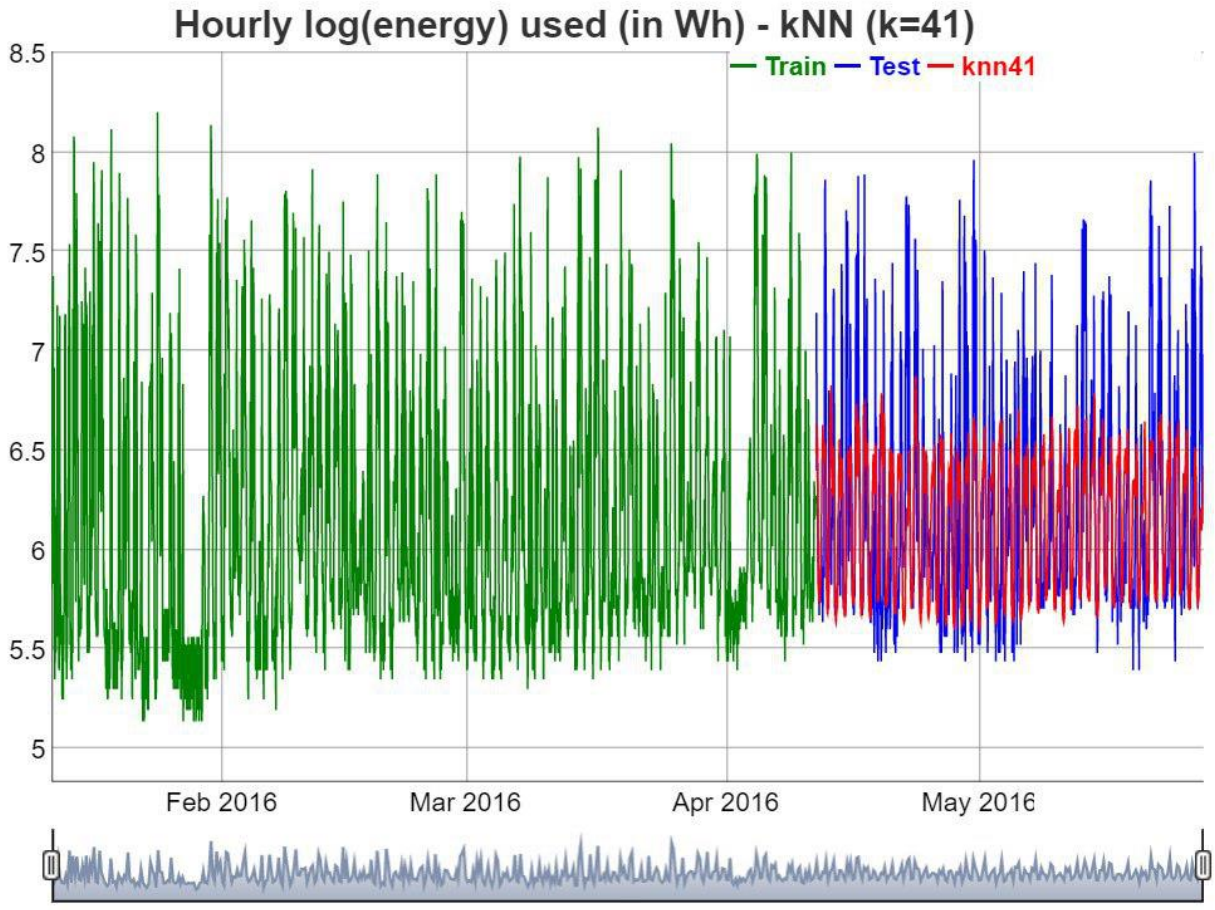

Figura 8: *previsioni del modello kNN (k=41)*

Con questo algoritmo si possono combinare modelli differenti: specificando un vettore nel parametro dei vicini, il risultato sarà un modello che ha come previsioni la media dei modelli con i numeri di vicini specificati nel vettore. Sono state provate diverse combinazioni per vedere se fosse possibile ottenere un singolo modello che potesse avere le caratteristiche dei kNN con pochi vicini, quindi che riesca a cogliere i picchi, ma un buon MAE come gli altri con più vicini. Il risultato non ha portato a delle migliorie: il MAE risultante era sempre intorno alla media aritmetica dei MAE dei modelli considerati singolarmente.

5. Confronto modelli

Come anticipato il confronto dei modelli viene effettuato valutando il MAE (trasformato per comodità) sul test.

Modello	**exp(MAE)**
UCM (doppia stagionalità)	213.06
ARIMA $(3,0,0)(0,1,1)$	219.33
kNN $(k = 41)$	220.50

Tabella 2: *confronto fra modelli in base al MAE*

L'errore commesso dal modello UCM è di 0.31, in valori trasformati 213.06 Wh all'ora. I due modelli migliori delle altre tecniche sono simili fra di loro e non lontanissimi dal migliore.

6. Conclusioni

Il modello migliore è risultato l'UCM con sia stagionalità giornaliera che settimanale. Questo risultato sembra ragionevole in quanto i consumi energetici variano sia all'interno di una giornata, sia all'interno della settimana in cui ci sono dei giorni dove il consumo è più elevato. I modelli di machine learning hanno buone performance se si sceglie il numero corretto di vicini, altrimenti possono essere molto lontani dall'ottimo. Il miglior modello ARIMA ha un MAE pressoché identico a quello del miglior kNN.

Di tutti i modelli costruiti si è sempre cercato di influenzarli per cogliere i picchi di consumo che altrimenti non erano mai predetti. Questa scelta si è rilevata vincente con il secondo modello UCM in cui l'aggiunta della stagionalità settimanale, anche se molto contenuta nei valori, ha portato ad un miglioramento delle previsioni. In altre situazioni come con i modelli di machine learning il voler a tutti i costi prevedere i picchi di consumo rischia di tralasciare le altre parti della serie.