Streaming Data Management & Time Series Analysis

AA 2022/2023

Analisi serie consumo di elettricità

Sanvito Simone 844794

Panoramica

Il progetto realizzato si focalizza sull'analisi, studio e previsione di una serie storica ad alta frequenza (dati ogni 10 minuti) riguardante i consumi energetici di una città del Marocco nel 2017. Sono stati utilizzati metodi di 3 famiglie, ARIMA, UCM e Machine Learning nella realizzazione del progetto e valutati tramite la metrica MAE.

Obiettivi

1. Predire i valori di power ogni 10 minuti per il periodo dal 01/12/2017 00:00:00 al 30/12/2017 23:50:00.
2. Utilizzare tre diversi metodi: un modello ARIMA, un UCM, e uno della famiglia Machine Learning.
3. Selezionare, tramite processi di validazione, gli algoritmi più promettenti, in modo da restringere la scelta ai "migliori" 3, uno per famiglia.

La bontà o meno di un modello è stata valutata tramite la metrica del MAE (Mean Absolute Error). Matematicamente rappresenta la distanza tra il valore predetto e quello effettivo.

Pre-processing

Il dataset è composto da 2 colonne: la data, che va dal 1 gennaio 2017 al 30 novembre 2017 in formato dd/mm/yyyy HH:MM:SS (formato giorno/mese/anno ora:minuti:secondi) e i valori sono rappresentati per intervalli di 10 minuti; la colonna y, che contiene informazioni sulla quantità totale di energia elettrica consumata in un determinato periodo di tempo.

In primis è stato effettuato un breve pre-processing: è stata controllata la presenza di possibili duplicati e valori mancanti nella serie storica e, non avendone trovati, si è continuato con la fase successiva.

Successivamente è stata realizzata una fase di data augmentation: infatti sono state aggiunte al dataset originale come colonne alcune features (come numero di giorno nell'anno o nella settimana) estratte dalla data fornita nel dataset originale.

Analisi Esplorativa e della serie

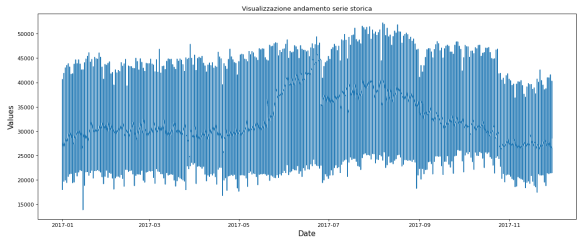

Figura 1: andamento della serie storica

In questo grafico si mostra l'andamento generale della serie nella sua interezza.

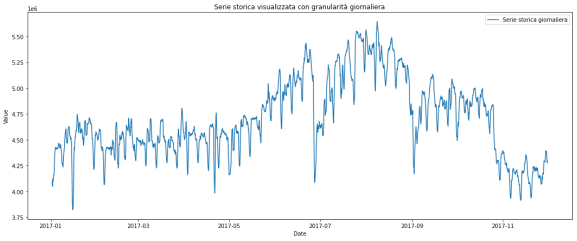

Figura 2: andamento della serie storica con granularità giornaliera

Tramite questi grafici risulta difficile trovare una tendenza, un trend evidente nei dati.

Effettuando ulteriori analisi si può vedere come all'interno di una settimana ci sia un andamento cumulato della serie, quindi un consumo, abbastanza simile, eccezion fatta per la domenica in cui si nota un abbassamento del livello di consumo.

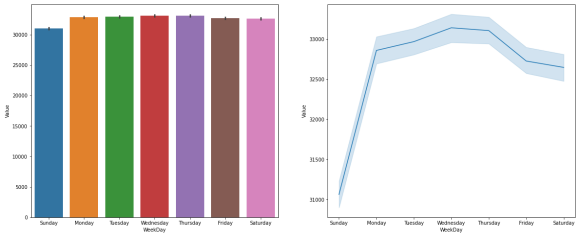

Figura 3: andamento della serie storica per giorni della settimana

Con granularità mensile, invece, si può notare come i mesi che hanno fatto registrare consumi cumulativamente maggiori sono quelli estivi, quindi giugno, luglio e agosto (e in parte settembre).

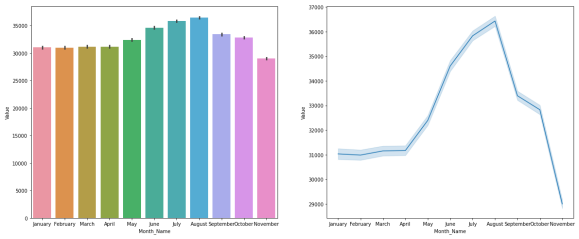

Figura 4: andamento della serie storica per mese

Successivamente sono stati analizzati i pattern di consumo orario, sia per giorno della settimana che per mese dell'anno.

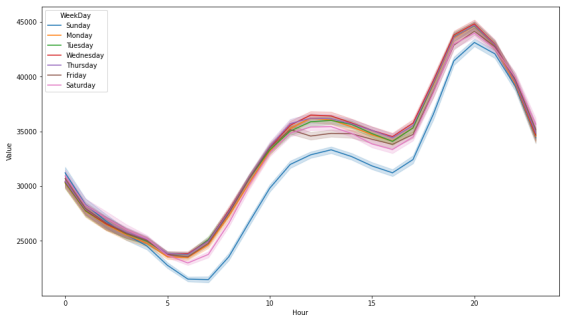

Figura 5: andamento della serie storica per ora divisa per giorno della settimana

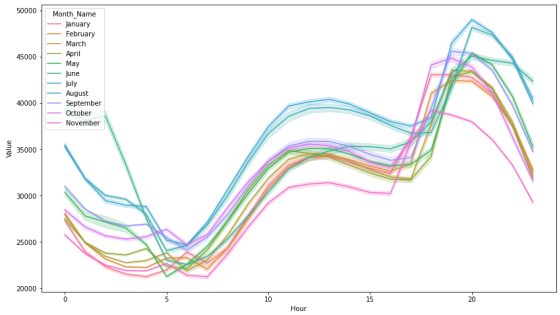

Figura 6: andamento della serie storica per ora divisa per mese dell'anno

Il consumo energetico durante il giorno varia in base a molteplici fattori, tra cui le attività svolte dalle persone, l'utilizzo di apparecchiature elettriche e la domanda generale. Tuttavia, in media, le ore con i consumi più bassi sono generalmente comprese tra le 4, le 5 e le 6 del mattino circa, quando la maggior parte delle persone è ancora a riposo e l'utilizzo di apparecchiature elettriche è minimo. D'altro canto, le ore con i consumi più elevati sono di solito comprese tra le 19 e le 20 di sera, quando le persone tornano a casa dal lavoro e accendono luci, televisori, computer, e altri dispositivi elettrici.

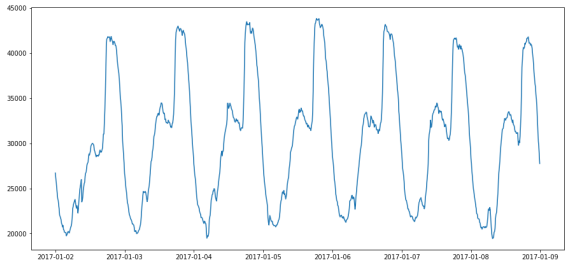

Figura 7: andamento della serie storica in una settimana specifica

Analizzando un caso specifico della serie in una settimana si nota una certa ripetitività del pattern giornaliero, e questo si può osservare per praticamente tutte le settimane dell'anno.

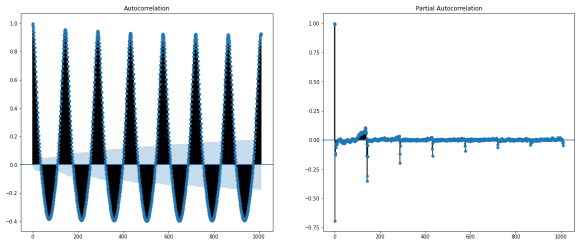

Figura 8: ACF e PACF della serie

Analizzando ACF e PACF della serie si nota come escano i ritardi ogni 144 osservazioni (e multipli). Questo va ad evidenziare la presenza di una stagionalità giornaliera (avendo infatti dati ogni 10 minuti si avranno 6*24 = 144 osservazioni ogni giorno).
Si può allo stesso modo osservare una stagionalità settimanale (quindi ogni 1008 osservazioni).

Si procede poi a realizzare un'analisi della stazionarietà della serie storica tramite i test ADF e KPSS.

L'Augmented Dickey-Fuller test (ADF) è un test statistico utilizzato per verificare la presenza di unit roots in una serie storica, ovvero per determinare se una serie temporale è stazionaria.
Il test di Dickey-Fuller (ADF) ha rilevato che la serie storica è stazionaria, rigettando l'ipotesi nulla di non-stazionarietà (il p-value è inferiore a 0,05). La serie storica è quindi considerata stazionaria.

Il Kwiatkowski-Phillips-Schmidt-Shin test (KPSS) è un altro test statistico utilizzato per verificare la stazionarietà di una serie temporale. A differenza dell'ADF, il KPSS testa l'ipotesi nulla che una serie temporale

sia stazionaria contro l'ipotesi alternativa che sia tendenzialmente instabile.
Il test di KPSS ha rigettato l'ipotesi nulla di stazionarietà, indicando che la serie storica non è stazionaria.

In questo caso, poiché i risultati dei due test differiscono (KPSS indica non stazionarietà e ADF indica stazionarietà) deve essere utilizzata la differenziazione per rendere la serie stazionaria. La serie può essere privata della tendenza attraverso la differenziazione o mediante l'adattamento del modello.
(Case 4: https://www.statsmodels.org/dev/examples/notebooks/generated/stationarity_detrending_adf_kpss.html).

Si è provato a rendere la serie stazionaria tramite la trasformazione logaritmica e tramite una Box-Cox, ma senza buoni risultati.

È stato provato quindi un altro approccio, tramite una differenziazione di 144 ordini.

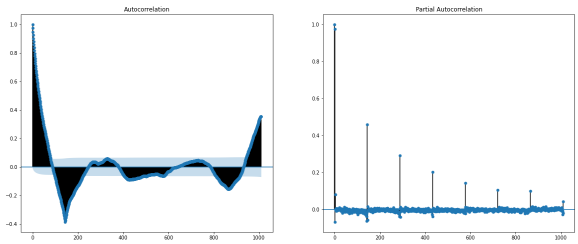

Figura 9: ACF e PACF della serie differenziata

Si riesce a risolvere la stazionarietà, ma resta comunque una correlazione a 144 osservazioni e ai suoi multipli, sottolineando la presenza di una stagionalità giornaliera e di una stagionalità settimanale. Dunque, non si è risolto il problema della stagionalità.

ARIMA

ARIMA (AutoRegressive Integrated Moving Average) è un modello statistico comunemente utilizzato nella previsione delle serie storiche perché combina la capacità di catturare relazioni tra i dati passati e di gestire la presenza di stagionalità e tendenze nelle serie temporali.

ARIMA utilizza una componente auto-regressiva (AR) per catturare la relazione tra un valore nelle serie e i valori precedenti, la media mobile (MA) per gestire la presenza di errori di previsione casuali, e la differenziazione integrata (I) per trattare la presenza di tendenze non stazionarie nelle serie.

Il dataset è stato suddiviso in due parti: il train set, che va dal 01/01/2017 00:00:00 fino al 31/10/2017 23:50:00, e il validation set che parte dal 01/11/2017 00:00:00 fino al 30/11/2017 23:50:00.

Come detto in precedenza, anche usando dei modelli ARIMA molto semplici con 1 differenziazione stagionale, la stagionalità giornaliera e settimanale non vengono eliminate.

Dunque si è deciso di procedere in altro modo, andando a costruire 72 serie storiche, aggregando per ogni 20 minuti di ogni ora (quindi 3 ogni ora, una per i minuti 0 e 10, una per i minuti 20 e 30 e una per i minuti 40 e 50).
In questo modo, prendendo dati di granularità inferiore alla granularità giornaliera, si può eliminare la stagionalità giornaliera e si può provare a modellare quella settimanale, mantenendo una differenziazione nella parte stagionale del modello.

Sono stati confrontati i modelli tra di loro calcolando il MAE sui dati aggregati, quindi è stato effettuato il confronto sulle predizioni aggregate rispetto al validation set aggregato.

ARIMA (1, 0, 1) (1, 1, 1) [7]

Questo modello è stato il primo considerato: restituisce un MAE di 2216.15.
Si è provato dunque a migliorare le predizioni.

ARIMA (1, 1, 0) (0, 1, 1) [7]

Proseguendo con la ricerca si è ottenuto un miglioramento del MAE con il modello ARIMA (1, 1, 0) (0, 1, 1) [7] che ha un MAE di 1001.58.
Il modello ARIMA (1, 1, 0) (0, 1, 1) [7] potrebbe funzionare meglio del modello ARIMA (1, 0, 1) (1, 1, 1) [7] poiché ha un termine I di ordine più elevato e un termine MA di ordine più basso, il che potrebbe consentire al modello di gestire meglio la tendenza e la stagionalità nei dati.

ARIMA (1, 1, 0) (0, 1, 2) [7]

Procedendo con la fase di fine tuning dei parametri, si è trovato un modello che avesse un MAE minore rispetto al modello precedente, con una memoria lineare simile a quella del modello precedente.
Dunque andando ad aumentare la componente di MA stagionale si ottiene un modello migliore, con MAE pari a 979.65.

ARIMA (2, 1, 0) (0, 1, 1) [7]

Questo è il modello che ottiene il MAE minore: 968.58.
Questo modello tiene conto di una maggiore quantità di informazioni sulle relazioni tra i valori passati rispetto al precedente.

Si sarebbero potute fare ulteriori esplorazioni, cercando di aumentare il valore dei parametri dei modelli per migliorare ulteriormente i risultati, ma a causa delle elevate necessità di tempo e computazionali richieste, mi sono fermato a questo modello, ritenendolo soddisfacente sia dal punto di vista del MAE trovato, sia per la pulizia dei residui.

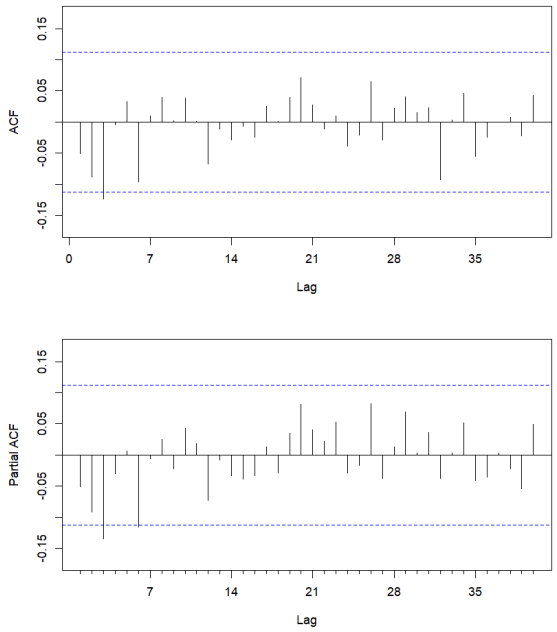

Figura 10 e 11: ACF e PACF di uno dei modelli costruiti con ARIMA (2, 1, 0) (0, 1, 1) [7]

Bisogna però riportare le predictions che determinano questo valore del MAE nel formato ogni 10 minuti, tramite l'operazione di spacchettamento descritta di seguito.

Spacchettamento in 10 minuti

I dati sono stati raggruppati per ora e numero di giorno dell'anno e sono state calcolate delle statistiche che sono state utilizzate come punto di partenza per l'operazione di spacchettamento. Infatti,

partendo dai dati aggregati, tramite delle funzioni ad hoc, è stata generata, tramite le statistiche calcolate, una coppia di valori (che avessero media pari al valore aggregato) che dividono i valori aggregati di ogni terzo di ora in due valori appunto. In particolare si verifica se la previsione successiva è maggiore o minore della previsione corrente e, in base a ciò, si ordina la sequenza in modo che i valori più grandi siano associati a previsioni successive più basse o viceversa.

In questo caso particolare, sono stati generati due valori per ogni 20 minuti: infatti sono stati spacchettati i dati su 3 dataframe diversi, uno per ogni terzo di un'ora, per poi unirli alla fine in un unico dataframe.

Dopo aver spacchettato i valori delle predictions ottenute tramite il modello ARIMA (2, 1, 0)(0, 1, 1)[7] il valore di MAE che si ottiene è 989.94.

In figura 12 si può vedere un confronto tra la serie originale e le predizioni ogni 10 minuti fornite dal modello ARIMA migliore.

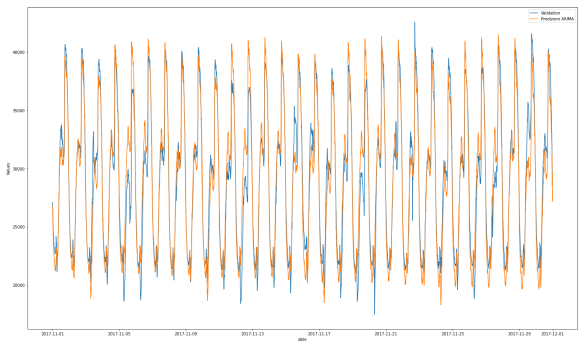

Figura 12: andamento del validation set vs predizioni ARIMA

UCM

L'altra famiglia di modelli che è stata utilizzata è quella dei modelli UCM (modelli a componenti non osservabili).

UCM (Unobserved Components Model) è un altro modello statistico comunemente utilizzato nella previsione delle serie storiche perché è in grado di catturare sia la componente di tendenza che la componente stagionale presenti nei dati.

UCM utilizza un approccio basato sulla decomposizione per identificare e trattare separatamente le diverse componenti presenti nei dati, come la tendenza, la stagionalità, la ciclicità e il rumore casuale. Questo modello è molto flessibile e adattabile, e può essere utilizzato in molti contesti diversi. Questo modello è in grado di fornire previsioni accurate e affidabili anche in presenza di dati con una complessa struttura.

Anche in questo caso è stata mantenuta la suddivisione del dataset in due parti: il train set, che va dal 01/01/2017 00:00:00 fino al 31/10/2017 23:50:00, e il validation set che parte dal 01/11/2017 00:00:00 fino al 30/11/2017 23:50:00.

Per quanto riguarda la parte di UCM, si è deciso di mantenere l'approccio delle 72 serie storiche, come fatto in ARIMA. Dunque non c'è stato bisogno di andare a modellare la stagionalità giornaliera; si è invece proceduto alla modellazione della stagionalità settimanale.

Sono stati considerati diversi tipi di livello e un diverso numero di armoniche (tra 1, minimo, e 10, massimo) di periodo 7, per modellare la stagionalità settimanale.

Dato che i modelli migliori sarebbero stati tutti con random walk come componente di livello, ma con numero di armoniche diverse, si è deciso di prendere il miglior modello per 3 tipi di livello diverso.
In base al livello selezionato il numero di armoniche migliore oscilla tra 8, 9 e 10; si sono presi 3 modelli, uno per un tipo di livello, che permettono di minimizzare il MAE del modello sul validation set.
I 3 migliori modelli trovati sono stati:

- "llevel", Local level con 10 armoniche;
- "rwalk", Random walk con 9 armoniche;
- "rwdrift", Random walk with drift con 8 armoniche.

Local Level con 10 armoniche

Con la componente di livello modellata come local level e l'uso di 10 sinusoidi per modellare la stagionalità settimanale si ottiene un MAE di 1231.16.

Random Walk con 9 armoniche

Con la componente di livello modellata come random walk e l'uso di 9 sinusoidi per modellare la stagionalità settimanale si ottiene un MAE di 1086.24.

Random Walk with Drift con 8 armoniche

Con la componente di livello modellata come random walk con drift e l'uso di 8 sinusoidi per modellare la stagionalità settimanale si ottiene un MAE di 1323.47.

Questi valori di MAE sono relativi ai dati che sono ancora "impacchettati", aggregati ogni 20 minuti. Per riportarli alla loro forma originale bisogna eseguire la fase di spacchettamento.

Spacchettamento in 10 minuti

La fase di spacchettamento per riportare le prediction nella forma originale, ovvero un'osservazione ogni 10 minuti, è stata uguale a quella descritta in ARIMA.
Quindi, spacchettando le predictions fatte con il modello migliore, il MAE passa da 1086.24 a 1106.92.

Quindi il modello migliore trovato, ovvero quello che minimizza il MAE sul validation set, è quello che modella la serie storica tramite un random walk utilizzando 9 armoniche per modellare la stagionalità settimanale della serie. Questo modello ha anche uno degli AIC più bassi, quindi riesce a spiegare in maniera efficiente i dati.

Di seguito si possono vedere gli andamenti della serie originale confrontata con le predizioni sul validation set per il modello UCM con random walk e 9 armoniche con periodo 7.

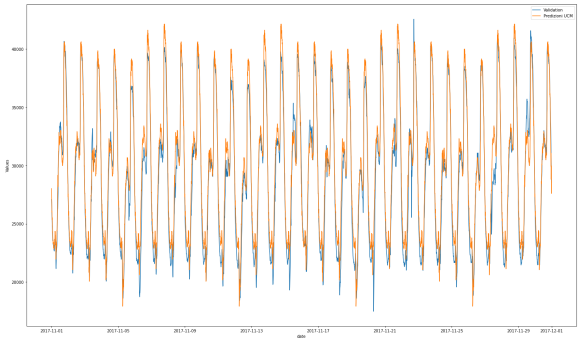

Figura 13: andamento del validation set vs predizioni UCM

Machine Learning

Per quanto riguarda il Machine Learning i modelli utilizzati sono stati tre: SVR, Random Forest e XGBoost.

In questo caso, per questioni di semplicità e di performance, è stata utilizzata la serie storica originale, senza aggregazioni, con train set che va dal 01/01/2017 00:00:00 al 31/10/2017 23:50:00, e il validation set che parte dal 01/11/2017 00:00:00 fino al 30/11/2017 23:50:00.

Inizialmente è stato provato un approccio con pochi regressori, ma questo non portava a risultati buoni. Quindi tramite un processo trial and error sono stati scelti i regressori da inserire all'interno dei modelli per ottenere dei MAE soddisfacenti.

Sono state inserite prima di tutto le features ricavate dalla data nella fase di preprocessing, come 'Num_DayofYear', 'Hour', 'Minute', 'Month', 'Num_DayofWeek', 'Num_WeekofYear'.

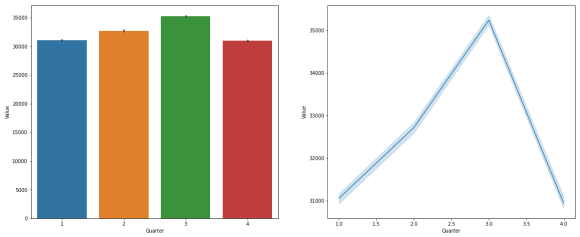

Figura 14: andamento della serie storica divisa per quarti di anno

È stata poi aggiunta la variabile 'Quarter': come si vede nella figura 14, il terzo quarto dell'anno è quello che ha fatto registrare i consumi più alti; inoltre, tramite questa variabile, dividendo l'anno in 4 periodi, si simulano anche le stagioni.

In seguito, data l'evidenza riportata in figura 5 e in figura 6, è stato creato un regressore per distinguere le ore notturne dalle ore diurne: le ore notturne vanno dalle 21 alle 5, quelle diurne dalle 6 alle 20.

Infine, essendo il dataset relativo ai consumi in Marocco, si è deciso di costruire un regressore per il periodo del Ramadan, periodo che dura dal 26/05/2017 00:00:00 al 24/06/2017 23:59:00.

Sono stati utilizzati diversi approcci, cercando la combinazione di iperparametri nei modelli che potessero minimizzare il MAE di ogni modello considerato.

SVR

SVR sta per Support Vector Regression, che è un algoritmo di regressione basato sulla tecnologia dei support vector machine (SVM). SVR viene utilizzato per fare previsioni di serie storiche perché è in grado di modellare relazioni non lineari tra le variabili di input e la variabile di output, ed è particolarmente utile per la previsione di dati che presentano relazioni complesse o sovrapposte. Inoltre, SVR è anche in grado di gestire dati con un gran numero di variabili e di modellare le relazioni non lineari tra le variabili di input e la variabile di output, rendendolo un'opzione efficace per la previsione di serie storiche.

Applicando questo tipo di modello dopo aver standardizzato i dati di input, sono state effettuate le previsioni sul validation set che restituiscono un MAE di 1991.07.

Sono stati provati quindi diversi approcci per vedere se si potessero ottenere performance migliori: mi sono focalizzato, soprattutto, su approcci basati su alberi decisionali, in particolare approcci gradient boosting e foreste.

Random Forest

Random Forest utilizza un insieme di alberi decisionali e valuta la maggioranza delle previsioni degli alberi per ottenere la previsione finale. Questo modello è molto flessibile e può gestire molteplici relazioni tra le variabili.

Sono stati implementati due metodi per trovare gli iperparametri ideali:

- Grid Search: consiste nel definire un insieme di valori per ciascun parametro e nel provare tutte le possibili combinazioni per trovare quella che produce i migliori risultati.
- Random Search: consiste nel generare valori casuali per ciascun parametro e nel provare combinazioni casuali per trovare quella che produce i migliori risultati.

In particolare la Random Forest con Grid Search ottiene un MAE sul validation set di 1570.77, mentre quella che cerca i valori migliori degli iperparametri tramite la Random Search restituisce un MAE sul validation set pari a 1335.57.

Questi algoritmi funzionano bene per la previsione dei consumi, poiché sono in grado di modellare relazioni complesse tra molteplici variabili e di gestire efficacemente dati con molte variabili o dati eterogenei. Inoltre, entrambi gli algoritmi sono altamente flessibili e possono essere adattati a molte situazioni diverse.

XGBoost

XGBoost utilizza una combinazione di alberi decisionali e ottimizzazione delle prestazioni per aumentare la precisione delle previsioni.

XGBoost è stato dimostrato essere molto efficace per molti tipi di dati ed è molto veloce rispetto ad altri algoritmi di apprendimento automatico; questo lo rende ideale per lavorare con grandi quantità di dati.

Per questo modello sono stati trovati gli iperparametri migliori tramite una procedura di Grid Search. In questo modo si ottiene un MAE di 1260.89, quindi un risultato migliore rispetto a SVR.

Quindi il modello di Machine Learning che fornisce le migliori previsioni è XGBoost.

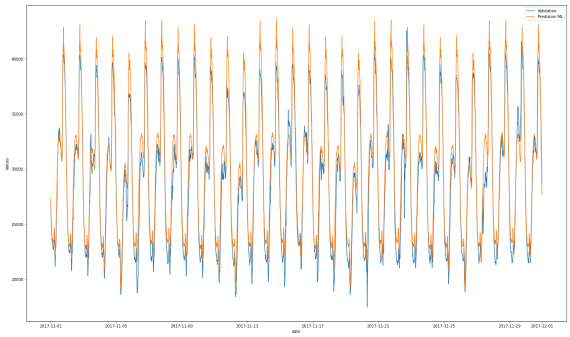

Figura 15: andamento del validation set vs predizioni ML

Ci sono diversi motivi per cui Random Forest e XGBoost potrebbero essere preferibili a SVR in termini di performance:

- Random Forest e XGBoost sono in grado di gestire le interazioni tra le variabili, che possono avere un impatto significativo sulla predizione della serie storica;
- Random Forest e XGBoost sono algoritmi di tipo ensamble che costruiscono una serie di modelli di decisione più semplici e poi li combinano per formare un modello più complesso e preciso. Questo li rende più robusti rispetto ai singoli modelli e meno sensibili ai casi outlier.

Predizioni su Dicembre

Per effettuare le previsioni (dal 01/12/2017 00:00:00 al 30/12/2017 23:50:00) sono stati utilizzati i modelli migliori per ogni famiglia:

- ARIMA (2,1,0)(0,1,1)[7] con train set dal 01/01/2017 00:00:00 al 30/11/2017 23:50:00; viene costruito aggregando i dati ogni 20 minuti e spacchettando successivamente le predizioni ogni 10 minuti.
  Sul validation set di Novembre ha un MAE di 989.94.
- UCM con random walk e 9 armoniche settimanali, con train set dal 01/09/2017 00:00:00 al 30/11/2017 23:50:00; viene costruito come ARIMA e spacchettato allo stesso modo.
  Sul validation set di Novembre ha un MAE di 1106.92.
- Machine Learning con XGBoost con Grid Search per trovare gli iperparametri migliori, con train set dal 01/01/2017 00:00:00 al 30/11/2017 23:50:00; usa la serie storica originale con l'aggiunta di alcuni regressori.
  Sul validation set di Novembre ha un MAE di 1260.89.

Di seguito vengono riportate le previsioni finali su Dicembre dei 3 modelli nello stesso grafico in modo da confrontarle tra di loro (ARIMA in blu, UCM in arancione, ML in verde).

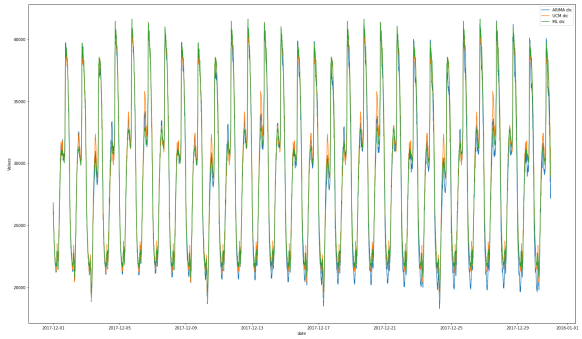

Figura 16: andamento delle predizioni su dicembre ARIMA vs UCM vs ML