ANALISI E PREVISIONE DEL CONSUMO ENERGETICO PER DUE EDIFICI DELL'UNIVERSITA' DI MILANO-BICOCCA

Riccardo Confalonieri
CdLM Data Science
matr. 830404
r.confalonieri5@campus.unimib.it

Rebecca Picarelli
CdLM Data Science
matr. 834286
r.picarelli1@campus.unimib.it

Silvia Ranieri
CdLM Data Science
matr. 878067
s.ranieri7@campus.unimib.it

Progetto di Data Science Lab
Appello del 24 Giugno 2021

Abstract

In questo lavoro si è cercato di analizzare l'andamento in termini di consumi energetici di due edifici specifici dell'Università Bicocca, ovvero l'U1 e l'U6, situati a circa 550 metri di distanza tra loro. Inoltre, con i dati a disposizione, riferiti in termini temporali agli anni tra il 2018 e 2020, è stato anche possibile ricavare informazioni sull'effetto del Covid a livello di andamento energetico di questa porzione di università. Tuttavia, sono state prese in considerazione, per meglio contestualizzare l'analisi e comprendere la natura dei risultati, le informazioni relative alle politiche di Bicocca volte alla risparmio energetico. In aggiunta, è stata effettuata un'integrazione con alcuni dati di natura meteorologica provenienti dalla stazione ARPA di Viale Marche. Ciò che è stato osservato complessivamente è stato un trend decrescente dei consumi energetici per l'U1 e un trend diametralmente opposto per l'U6. Inoltre l'algoritmo di machine learning Random Forest si è rivelato essere il più performante, rispetto ad altre tecniche ed algoritmi inerenti all'ambito delle serie storiche, in termini di previsione dei valori anche su un intero anno.

Keywords

Consumi energetici; Predizione; ARIMA; SARIMA; Machine Learning

Indice

1 Introduzione

La tematica dei consumi energetici, elettrici e termici e la relativa gestione, è sempre stata un argomento importante per l'Ateneo in termini sia ambientali che economici. La sostenibilità energetica richiede diverse attività quali: il monitoraggio, il riconoscimento delle inefficienze, l'ottimizzazione e l'incentivo al risparmio energetico. Dunque le azioni dovranno focalizzarsi oltre che sulla gestione degli impianti anche sull'incentivare una buona condotta verso tutto il personale scolastico.L'Area Infrastrutture e Approvvigionamenti a cui è affidato il piano di gestione delle utenze e dei consumi energetici ha l'obiettivo di promuovere azioni strutturali-gestionali ed educativo-comportamentali il cui scopo è risparmio ed efficienza energetica. Nel corso degli anni, ad esempio, sono state effettuate diverse attività di rilevazione ed indagine dei consumi energetici di tutti gli edifici dell'Ateneo con il fine di identificare le criticità e ideare eventuali piani di azione mirati ad ottenere dei miglioramenti nel tempo.
Alla luce di tutto ciò, nello svolgimento di questo lavoro e nella contestualizzazione dei risultati, è stata data particolare rilevanza ad alcune informazioni rilasciate dall'Ateneo [1], tra le quali si può citare come il consumo energetico riferito al singolo utente sia in leggera diminuzione sebbene vi sia una crescita complessiva in termini di studenti ed edifici, in aggiunta si riporta che l'università non riscontra alcun incremento nella spesa energetica nel corso degli anni. Inoltre un altro aspetto molto rilevante nelle considerazioni fatto è stata la forte politica di efficientamento energetico seguita da Bicocca, importanti investimenti sono stati fatti sull'illuminazione degli edifici e agli impianti di condizionamento. Più precisamente, è stato completato il piano di sostituzione dell'illuminazione degli spazi comuni di tutti gli edifici con lampade LED, con un risparmio in termini economici rispetto ai sistemi di illuminazione tradizionali (per lo più lampade alogene) che si avvicina al 90%. In termini di consumi, i LED permettono infatti di risparmiare in media il 38% di energia a parità di illuminazione offerta. Per quanto riguarda il sistema di raffrescamento per il periodo estivo, sono stati installati nuovi e più efficienti impianti di produzione di energia frigorifera. I nuovi apparecchi si caratterizzano per un'altissima efficienza energetica e un basso impatto acustico. Inoltre, sono dotati di un software per la gestione in remoto che permette un controllo costante dell'attività e un'ottimizzazione dei consumi energetici che si sono così ridotti del 42% con una contestuale riduzione delle emissioni prodotte pari a 390 tonnellate di CO2 l'anno. Queste politiche risultano ancora più importanti se si pensa che il numero di aule ed edifici di Bicocca è in continua crescita negli anni.

2 Obiettivo

L'obiettivo del progetto è quello di prevedere l'andamento dei consumi energetici nel tempo, per definire un piano di gestione degli impianti e rendere ancora più efficiente il sistema di monitoraggio. Inoltre le predizioni dell'andamento, se accurate, permetterebbero di programmare eventuali interventi riducendo al minimo i disagi e gli inconvenienti sia per gli studenti sia per il personale universitario.
Quest'analisi è dunque rilevante sia per i soggetti interessati e destinatari ovvero gli operatori dell'area infrastrutture e approvvigionamenti dell'Università di Bicocca, a cui è affidato la direzione e il controllo dei consumi, sia per gli studenti per attività di ricerca applicativa nell'ambito della sostenibilità ambientale.

3 Aspetti metodologici

In questo lavoro è stato deciso di implementare diversi approcci e strategie al fine di effettuare le previsioni dei consumi energetici, tali metodologie sono state implementate dopo aver effettuato delle analisi preliminari pertinenti con la natura dei dati inoltre sono state considerate diverse finestre temporali per la predizione. Infine nella scelta finale si sono considerati anche i risultati ottenuti dai modelli utilizzati per verificarne l'attendibilità delle predizioni. Inoltre, si precisa che per stabilire il valore dei parametri richiesti dai modelli sono state prese in considerazione informazioni rintracciate nella letteratura inerente a questo settore, anche se talvolta le scelte sono semplicemente scaturite da valutazioni di natura più empirica.
Più precisamente, comunque, sono state costruite diverse versioni del modello autoregressivo integrato a media mobile (ARIMA), inclusa quella che prevede un approccio definito "rolling", ovvero facendo uso di finestre mobili, utile quando il classico ARIMA non riesce a prevedere correttamente diversi istanti temporali in avanti. In questo caso la predizione è fatta come $t + h|t$ dove h è l'ampiezza di tempo considerata per il forecast mentre t è il giorno di partenza della previsione, dunque i dati sul quale è fittato il modello arriveranno proprio fino al t-esimo giorno. Questo approccio ha permesso di ottenere previsioni su più mesi anche con pochi dati a disposizione. Inoltre la gestione della componente stagionale, per la quale solitamente si implementa il SARIMA, è risultata non implementabile in alcune casistiche poi descritte per via della mancanza di un numero minimo di osservazio-

ni necessarie per il fitting. In aggiunta, si ha optato per inserire nell'analisi anche alcune strategie provenienti dal mondo del machine learning e deep learning, ovvero l'algoritmo delle foreste casuali e la rete neurale ricorrente di tipo Long short-term memory. In entrambe queste due ultime soluzioni, è stato deciso di procedere impostando un'analisi di previsione di tipo supervisionato, inserendo a seconda dei casi una sola variabile, ovvero quella chiave del consumo energetico, oppure associandola ad altre di natura meteorologica, ossia quelle di temperatura, umidità e vento rilevate nei pressi degli edifici dell'Ateneo di Bicocca.

4 I dati

I dati utilizzati sono stati forniti dall'Università degli studi di Milano-Bicocca e riguardano il consumo di energia relativo agli edifici U1 e U6 per gli anni 2018-2019-2020. Nel dataset erano presenti diverse variabili, tra cui quelle di interesse sono:

- **Data:** riporta la data della in formato stringa.
- **Ora:** riporta l'ora della rilevazione in formato stringa.
- **Consumo attiva prelevata:** è l'effettivo consumo di energia rilevato in kw (potenza).

Il dataset riportava queste informazioni per entrambi gli edifici con rilevazioni continue ogni 15 minuti, per ipotesi è stato assunto che il consumo fosse uniforme per l'intero intervallo di tempo tra una rilevazione e l'altra.
Inizialmente si è dovuto effettuare delle operazioni di data cleaning per poter utilizzare correttamente i dati, i dati infatti erano riportati in file di formato diverso e alcuni mesi erano divisi su più file diversi quindi si è proceduto a normalizzare il formato e creare dei file unici per i diversi mesi dell'anno, mantenendo comunque la suddivisione tra gli edifici. Corretti questi errori nei file si è proceduto alla lettura e all'elaborazione degli stessi tramite Python [2], i dati sono stati quindi preparati per l'elaborazione formattando in modo appropriato i campi "Data" e "Ora" ed eliminando le letture duplicate presenti in nei diversi file.
Infine è stato necessario, ai fini delle analisi, calcolare l'effettivo consumo energetico in kWh per l'intera giornata invece che per ogni 15 minuti, per farlo si è calcolata la media del consumo orario e successivamente sommando i consumi energetici delle singole ore della giornata.
Ripuliti entrambi i dataset si è proceduto studiando più nel dettaglio i valori a nostra disposizione attraverso una serie di grafici e di analisi. Si sono dunque studiati:

- L'andamento dei consumi nei vari giorni della settimana.
- L'andamento dei consumi nei vari mesi dell'anno.
- L'andamento dei consumi nei diversi anni a disposizione.
- L'andamento dei consumi nelle diverse fasce orarie.

Questi grafici, per l'edificio U1, hanno permesso di evidenziare che nel weekend si osserva una leggera diminuzione dei consumi soprattutto per la domenica. Questo perché di sabato solitamente gli edifici sono aperti per spazi studio e biblioteche e quindi si spiega il minor decremento rispetto alla domenica. A livello mensile si registra invece un forte aumento dei consumi nei mesi estivi, in quanto U1 utilizza l'energia anche per il raffrescamento degli spazi e quindi era un risultato abbastanza atteso. Mentre per quanto riguarda i diversi anni si è registrato un costante decremento dei consumi sia tra 2018 e 2019 (-9.07%) ma anche tra 2019 e 2020 (-6.22%), questo può essere dovuto a molteplici fattori, soprattutto per il 2020, come ad esempio politiche di efficientamento energetico attuate da Bicocca o anche un minor utilizzo dovuto alla chiusura per il Covid-19. Non avendo ulteriori informazioni è però difficile trarre conclusioni e quindi anche nell'utilizzo dei modelli si è provato a considerare entrambe le casistiche escludendo prima il 2020 dalle analisi e includendolo solo in seconda battuta. L'analisi per fasce orarie ha fatto emergere come il consumo aumenti tra le 11 e le 15, probabilmente per l'utilizzo degli studenti e del personale di microonde e di altre apparecchiature per il pranzo. Si riporta in figura 1 il grafico dell'andamento orario dei consumi.

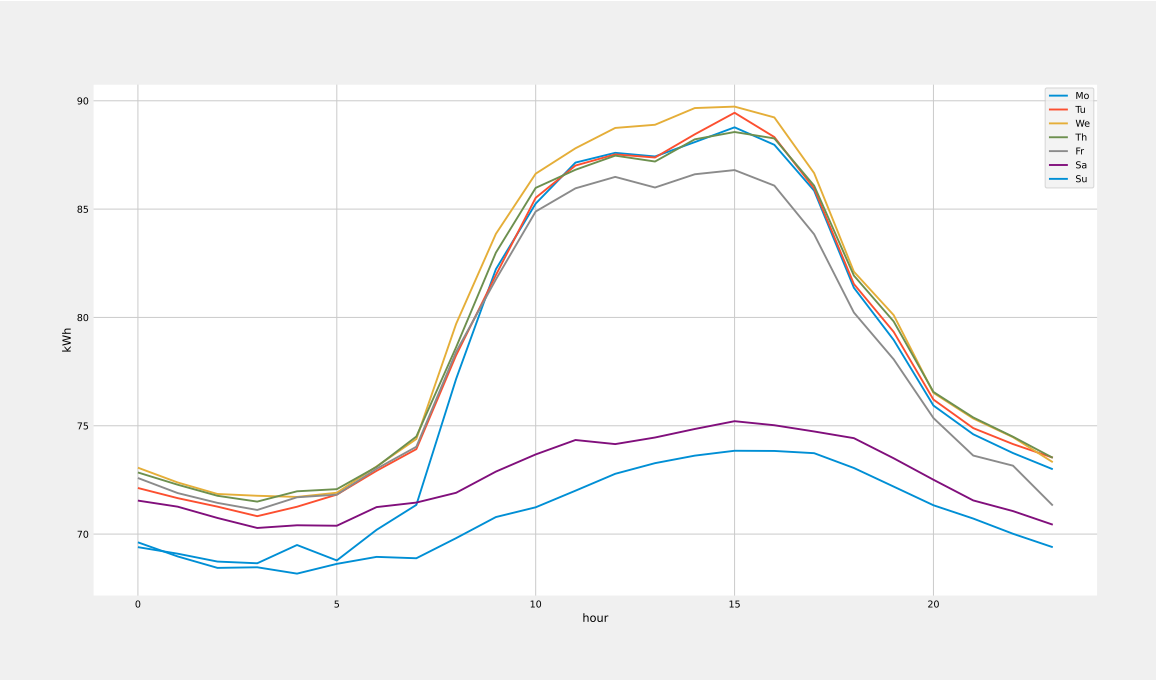

Figura 1: Andamento orario dei consumi per l'edificio U1

Per l'edificio U6 molte delle osservazioni appena fatte rimangono valide, sebbene non si nota un incremento dei consumi nei mesi estivi in quanto in questo edificio il raffrescamento utilizza altre fonti di energia. Inoltre a livello annuale la situazione è diversa rispetto all'edificio U1, si osserva infatti un incremento dei consumi tra 2018 e 2019 mentre tra 2019 e 2020 si evidenzia un decremento abbastanza importante che, in questo caso, è probabilmente dovuto all'impatto del Covid-19. Per quanto riguarda i consumi nelle fasce orarie rimangono molto simili all'altro edificio anche se il sabato si avvicina ancora di più ai consumi infrasettimanali in quanto in U6 vi è la presenza della biblioteca e alcuni laboratori e aule sono utilizzati per lezioni anche il sabato a differenza di U1.
Al termine di queste analisi preliminari ci si è concentrati sullo studio delle rilevazioni con valori anomali, si sono considerati come anomalie quelle registrazioni al di fuori del range $[\mu - 2\sigma, \mu + 2\sigma]$. Per entrambi gli edifici si sono registrati valori più bassi nella soglia soltanto in giorni di vacanza (ad esempio: Natale, Pasqua) o nei giorni immediatamente prima e dopo quando la festività cadeva ad esempio di giovedì quando da calendari didattici era previsto il ponte a favore degli studenti. Mentre i valori più alti della soglia sono differenti per i due edifici dato il diverso utilizzo dell'energia, in U1 si sono infatti registrati valori particolarmente alti per periodi estivi, dunque di maggior caldo, quando ancora l'università non era chiusa e dunque vi era la necessità del raffrescamento. Mentre in U6 i valori più alti si sono registrati nei mesi invernali il che fa pensare che l'energia sia in qualche modo collegata all'utilizzo del riscaldamento.
Si sono poi rilevati dei casi in cui le registrazioni hanno segnalato 0 come valore di consumo, in entrambi casi la giornata del 31-07-2020 riporta come valore zero per tutto l'arco della giornata e questo fa dedurre che vi siano stati errori di rilevazione, cercando alcuni riferimenti storici si è infatti scoperto che in quella giornata sono state effettuate diverse segnalazioni di blackout in tutta Milano. L'edificio U1 riporta come zeri anche a mezzanotte del 29/06/2019, quando nuovamente erano stati segnalati vari blackout, e alle ore 10 del 12/07/2019. Mentre l'edificio U6 ha come valore zero in alcuni valori orari dei giorni 27/12/2019, 11/04/2020 e 25/10/2020 ma questi zeri sono stati registrati per brevissimi intervalli di tempo e ciò fa pensare ad errori nella lettura dei consumi. Per tutti questi giorni segnalati Bicocca, solita a mandare mail di avviso agli utenti, non ha segnalato alcun intervento programmato.
Oltre all'utilizzo dei valori puntuali registrati sono stati utilizzati anche altri valori come covariate nei modelli, "Weekday" che riporta il numero di giorno della settimana. Mentre in altri modelli i dati sono stati integrati con dei dati meteo, ottenuti tramite una richiesta all'ARPA(Agenzia regionale per la protezione ambientale), relativi alle variabili vento, temperatura, umidità rilevate dalla stazione viale Marche situata a circa 3km dall'università

Bicocca. Queste variabili hanno permesso di studiare maggiormente l'andamento dei consumi e in alcuni casi l'impatto sui modelli è stato positivo come verrà poi descritto, soprattutto per l'edificio U1. Nell'appendice A, tabelle 3 e 4, sono riportati i valori di correlazione ottenuti per le variabili considerate per entrambi gli edifici.

5 Edificio U1: Analisi effettuate

Come descritto precedentemente l'obiettivo del progetto è di riuscire a prevedere l'andamento della serie storica nel futuro, per questo sono stati implementati diversi modelli per prevedere il consumo di energia. Date le diversità tra i due edifici le analisi effettuate sono state diverse e quindi verranno descritte singolarmente, nei prossimi paragrafi ci si concentrerà sull'edificio U1.
Al fine di costruire i modelli migliori ci si è concentrati inizialmente sullo studio della struttura della serie storica attraverso la decomposizione spettrale che, come si può notare dalla figura 2, risulta fortemente stagionale e, come già si poteva parzialmente intuire dallo studio iniziale sui dati, con un trend decrescente. Date queste premesse abbiamo verificato, attraverso il test di Dickey-Fuller, la stazionarietà della serie che risulta soddisfatta soltanto dopo averla differenziata di un lag. Questo procedimento ha portato a definire il parametro $d = 1$ valido per i modelli che successivamente sono stati applicati.

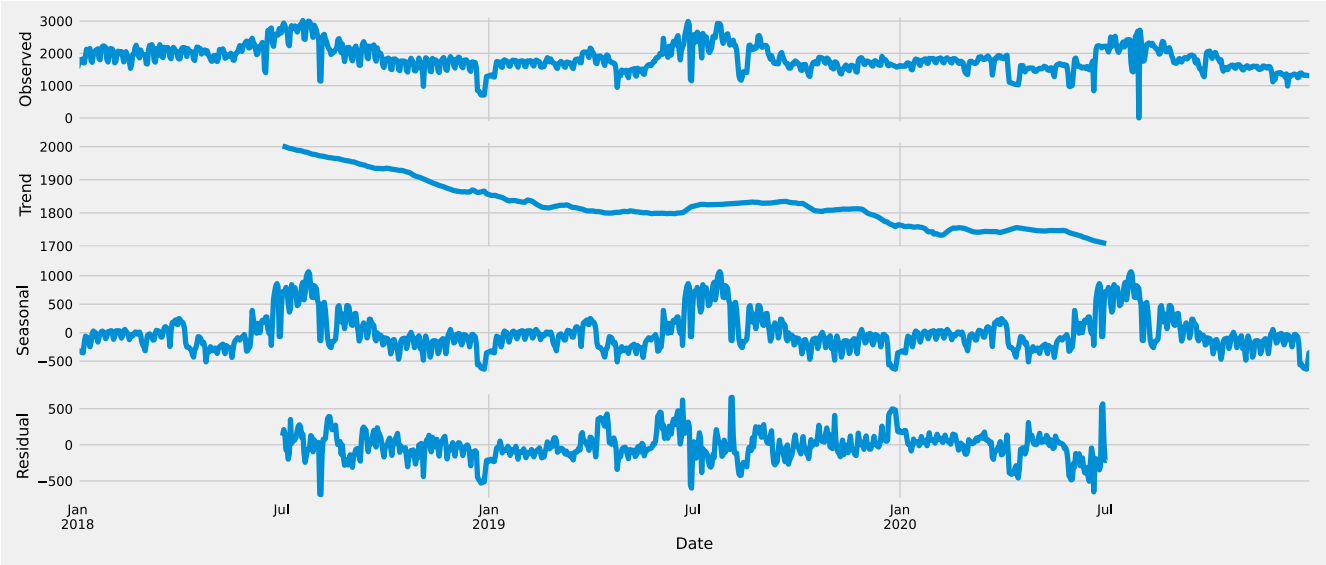

Figura 2: Decomposizione spettrale della serie storica per l'edificio U1

Per la scelta degli altri parametri p e q si è utilizzato un approccio automatico che ha permesso di trovare la miglior combinazione secondo il criterio della minimizzazione del AIC, tale approccio è disponibile attraverso la libreria python [3] che mette a disposizione il metodo `auto_arima`. Tuttavia questo approccio richiede il fitting di diversi modelli per tutte le possibili combinazioni, per ridurre in modo sensato i tentativi si è analizzato i grafici dell'autocorrelazione (ACF) e dell'autocorrelazione parziale (PACF) che hanno permesso di ridurre il numero di tentativi. Al termine di quest'analisi si sono dunque ottenuti i valori dei parametri necessari per fittare il modello ARIMA che in questo caso equivalgono a `p = 7, d = 1, q = 8`.
Ottenuti i parametri ideali si è potuto fittare il modello con i dati a disposizione, in particolare sono stati realizzati diversi modelli più o meno efficienti, nei prossimi paragrafi verranno illustrati nel dettaglio quelli maggiormente significativi. In particolare si riportano sia i modelli con i soli anni 2018-2019 sia i modelli che includono nello studio anche il 2020 in base a quanto descritto nel paragrafo 3.

5.1 Rolling ARIMA (7,1,8) anni 2018-2019

Dato che gli anni a disposizione non erano sufficienti per utilizzare i modelli SARIMA si è deciso di cambiare metodologia per affrontare il problema, si è dunque passati ad un approccio "rolling". Ovvero, una volta fittato il modello vengono predetti i valori dei 5 giorni successivi, dopo di che si ri-fitta il modello includendo nel dataset

anche i valori reali dei 5 giorni successivi. In questo modo il modello permette di prevedere sempre 5 giorni in avanti sebbene sia necessario ri-fittarlo step-by-step. Questo approccio si è reso necessario in quanto il modello ARIMA non interpolava correttamente la stagionalità della serie e dunque su finestre ampie di predizione tendeva a prevedere un valore costante che non era in alcun modo collegato con il reale andamento dei consumi. Per la realizzazione del modello in python si è utilizzata la libreria [4], per l'implementazione del codice vedesi appendice B, source code 1.
La finestra di predizione può essere aumentata o diminuita cambiando il valore dell'apposita variabile `num_steps_pred`, in particolare dopo vari tentativi si è visto che 5 giorni risulta essere la finestra più adeguata per predire valori il più possibile realistici.
Una *variante* di questa implementazione prevede invece il fitting del modello inserendo anche la covariata "Weekday", riportante il giorno della settimana in formato numerico, che permette di aumentare leggermente la finestra di qualche giorno mantenendo comunque buone performance predittive. Vedesi appendice B, source code 2, per l'implementazione python.

5.2 Risultati Rolling ARIMA (7,1,8)

Il modello fittato seguendo gli step appena descritti risulta essere valido dal punto di vista statistico, infatti dai grafici di diagnostica, riportati in figura 3, risultano correlazioni all'interno della banda di normalità e un buon adattamento alla distribuzione normale anche se nelle code qualche osservazione tende ad allontanarsi

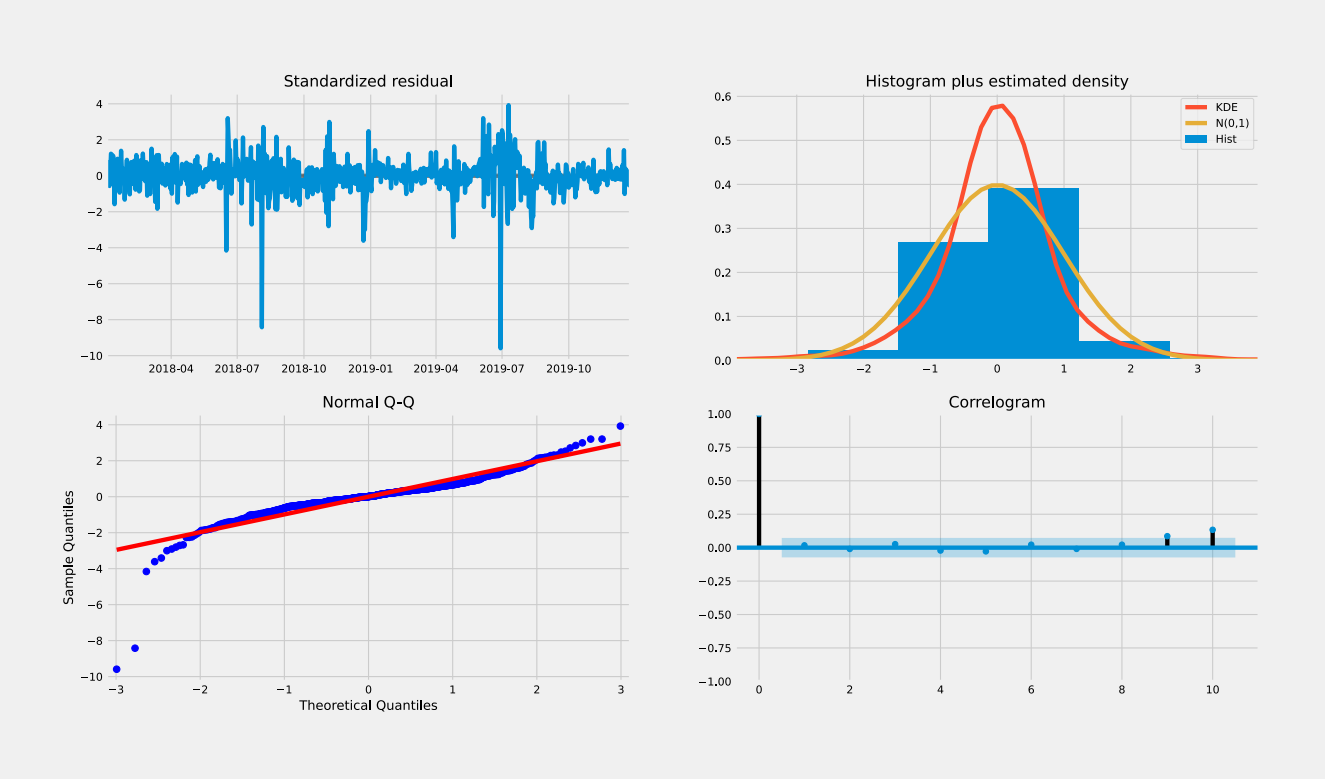

Figura 3: Rolling ARIMA (7,1,8) a 5 giorni: Grafici di diagnostica U1

Oltre a valutare la bontà del modello si è anche valutata l'accuratezza delle previsioni, si è dunque calcolato l'indice R^2 tra i valori predetti dal modello e i valori realmente osservati sfruttando il metodo apposito presente nella libreria [5]. Nel caso della finestra a cinque giorni si è ottenuto un R^2 pari a 0.51, un risultato buono che garantisce una previsione piuttosto accurata. In figura 4 è possibile confrontare graficamente l'andamento dei due valori, inoltre in figura è rappresentato l'intervallo di confidenza standard al 95% anche se in questo caso la confidenza dei valori predetti rispetto ai reali è molto più accurata.

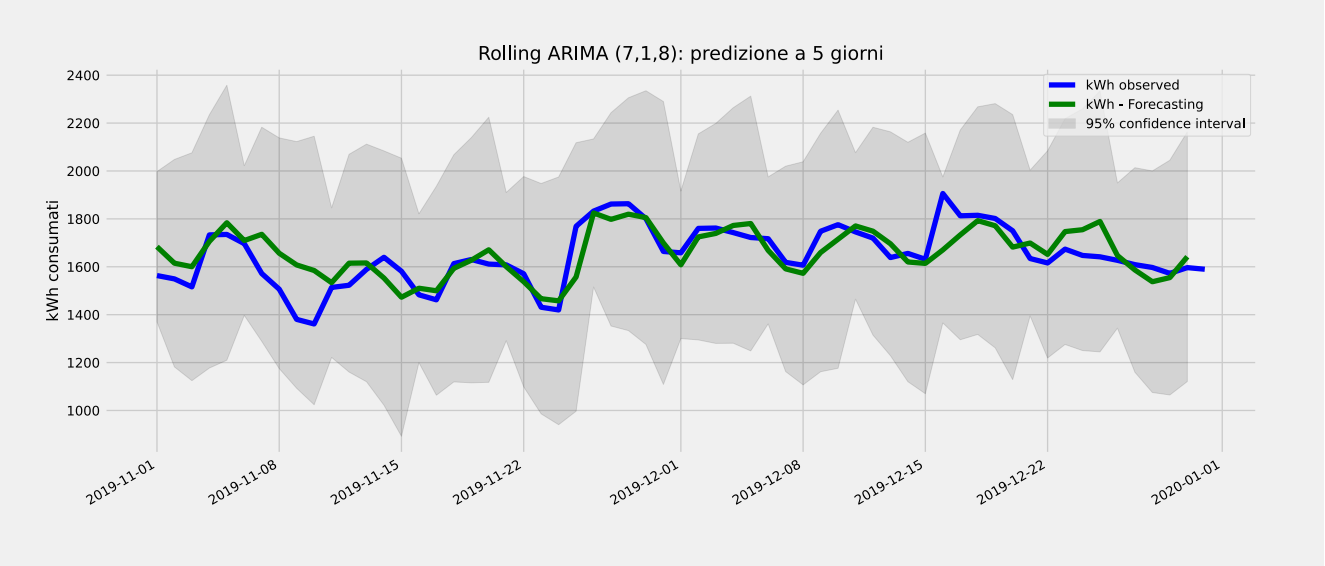

Figura 4: Rolling ARIMA (7,1,8) a 5 giorni: valori predetti fine 2019 U1

Verificato che la predizione risultasse abbastanza buona si è provato ad utilizzare l'algoritmo appena enunciato anche per prevedere l'andamento nel 2020, sempre utilizzando una finestra mobile a 5 giorni, utilizzando in questo caso il biennio 18-19 per il fitting del modello mentre tutto l'anno 2020 è stato utilizzato come test set. Questo perché, per quanto detto precedentemente, non si può escludere a priori che il decremento registrato nel consumo sia dovuto al covid-19 e non a politiche di efficientamento energetico. Dai risultati si evince che questa metodologia risulta valida anche su lunghi periodi, l'indice R^2 ottenuto con una previsione a 7 giorni per tutto l'anno 2020 si attesta infatti a 0.430. Un riscontro completo sui valori ottenuti per l'indice R^2 è consultabile nel appendice A, Tabella 1.
Si ritiene che questi risultati possano essere d'aiuto per monitorare il consumo elettrico previsto nei giorni immediatamente successivi. Ad esempio permette di attuare eventuali politiche di risparmio energetico, laddove ve ne sia bisogno, evitando di mantenere i consumi particolarmente elevati per più giorni consecutivi e più in generale permette al personale preposto di verificare eventuali giornate anomale che richiedono un consumo molto più elevato rispetto a quello inizialmente previsto dal modello garantendo così verifiche più accurate ed interventi più tempestivi per mitigare l'impatto sui costi.

5.3 SARIMA anni 2018-2020

Con l'introduzione nelle analisi dell'annata 2020 è stato possibile utilizzare i modelli SARIMA, ovvero includere nel modello anche la stagionalità della serie storica. Tuttavia in questo caso non è stato possibile utilizzare Python in quanto la frequenza giornaliera dei dati richiedeva troppo tempo per essere elaborata, il problema è stato risolto utilizzando R [6] che in questo caso è risultato più efficiente e ha permesso di fittare il modello senza ulteriori manipolazioni sulla granularità dei dati.
Per l'implementazione del codice si sono utilizzate le librerie TSeries [7] e Forecasting [8], che hanno permesso rispettivamente di estrapolare la serie storica del consumo dal dataset e di fittare correttamente il modello. Anche in questo caso, come in precedenza, si è utilizzata la funzione `auto.arima` offerta dal secondo pacchetto software per trovare i parametri corretti per la serie secondo il medesimo criterio di minimizzazione dell'AIC. In particolare i parametri in questo caso sono risultati essere `order = (7,1,8), seasonal = (0, 1, 0, 365)`, è stato fatto un tentativo anche con i parametri `order = (5,1,2), seasonal = (0, 1, 0, 365)` con i quali si ottiene un AIC e risultati in termini di forecasting simili ma con un tempo di fitting notevolmente minore, tuttavia è stato preferito mantenere la prima configurazione per via dei residui migliori ottenuti dal modello come poi verrà descritto nel prossimo paragrafo. Una volta individuati i parametri più corretti si è proceduto con il fitting del modello e la predizione dei valori, in questo caso per il fitting sono stati usati tutti i valori da Gennaio 2018 a Ottobre 2020 mentre i restanti due mesi del 2020 sono stati riservati per la verifica dell'accuratezza delle predizioni. Vedesi appendice B, source code 4, per l'implementazione del codice in R.

5.4 Risultati SARIMA

Il modello SARIMA con `order = (7,1,8), seasonal = (0, 1, 0, 365)` ha richiesto circa 50 minuti di esecuzione per fittare tutti i valori del dataset in input, fino a ottobre 2020 in questo caso. Tuttavia il modello finale è risultato abbastanza buono dal punto di vista statistico, come si può verificare in figura 5 i residui infatti rientrano tutti nella banda di normalità fatto salvo per due lag molto elevati, 17 e 23. Il modello, rispetto a quelli ottenuti con arima, si discosta invece maggiormente dall'andamento della normale soprattutto nelle code che però risultano simmetriche e dunque riteniamo il risultato comunque accettabile.

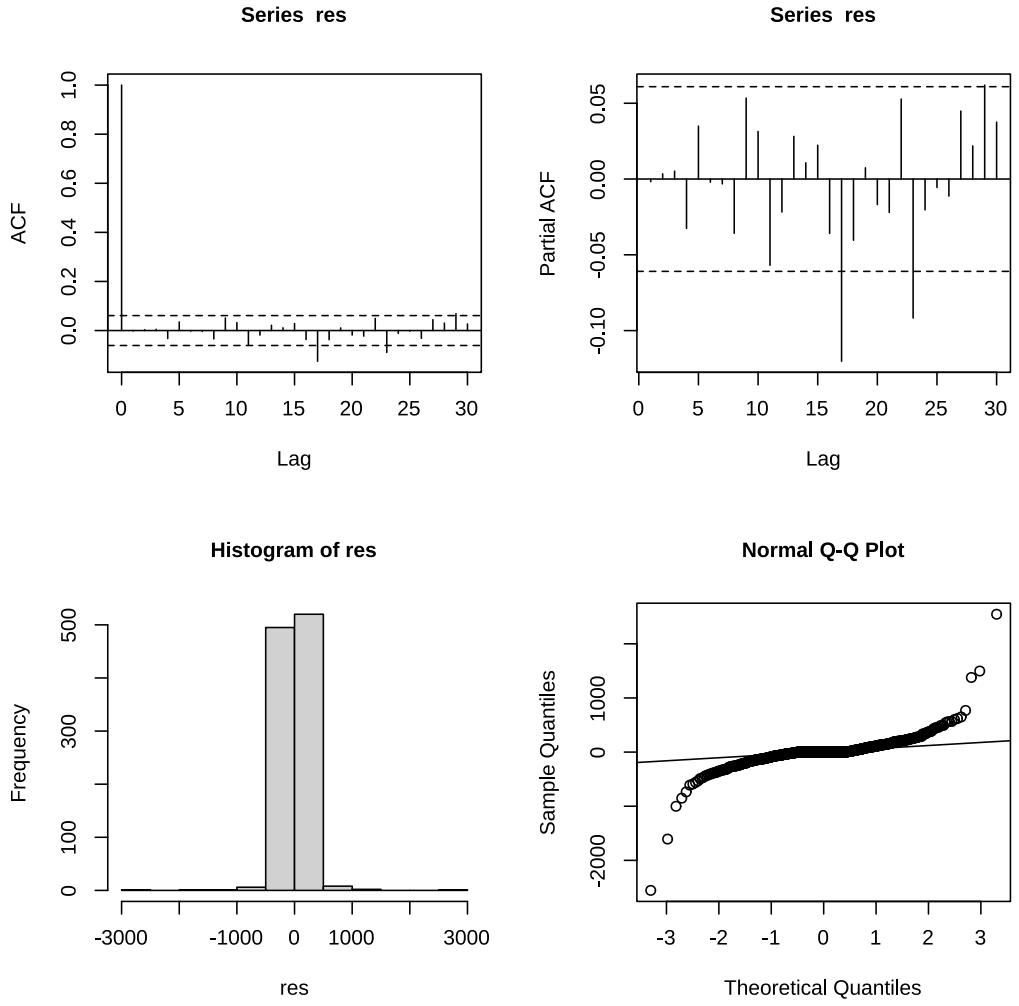

Figura 5: SARIMA (7,1,8) (0,1,0,365): Grafici di diagnostica per U1

Dal punto di vista della predizione si sono valutati, come prima, i valori predetti rispetto ai valori reali osservati negli ultimi due mesi del 2020 calcolando l'indice R^2 che riporta come score 0.23. Tale valore non è particolarmente alto anche se rimane comunque uno score positivo e permette di prevedere una finestra più ampia di due mesi, in figura 6 e 7 si può notare a destra il risultato della predizione rispetto alla serie storica utilizzata per il fitting, mentre la figura a sinistra riporta lo zoom rispetto al solo periodo dei valori predetti.

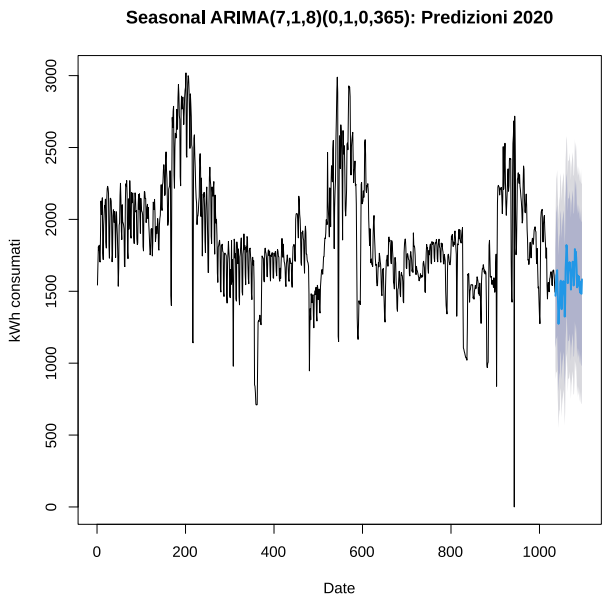

Figura 6: SARIMA (7,1,8) (0,1,0,365): valori predetti

In figura è riportata in nero la serie storica conosciuta, in blu i valori predetti e, rispettivamente in grigio chiaro e scuro sono riportati gli intervalli di confidenza al 95% e al 80%

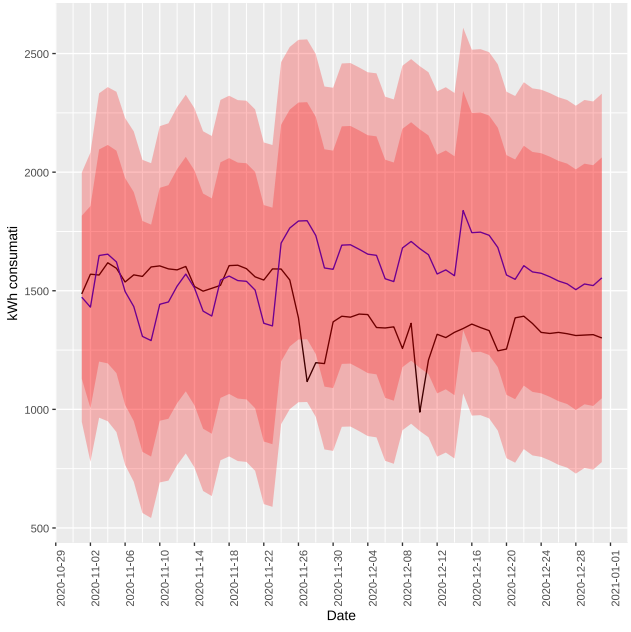

Figura 7: SARIMA (7,1,8) (0,1,0,365): zoom valori predetti

In figura sono riportati in nero i valori realmente osservati, in blu i valori predetti nero e, rispettivamente in rosso chiaro e scuro sono riportati gli intervalli di confidenza al 95% e al 80%

Analizzando più attentamente la figura si nota che la previsione è molto più vicina ai valori reali per i primi 22 giorni, successivamente tende un po ad allontanarsi prevedendo valori più elevati rispetto alla realtà soprattutto in corrispondenza dei due picchi di fine novembre e inizio dicembre. Tuttavia tutti i valori predetti rientrano nell'intervallo di confidenza all'80% fatta eccezione per i giorni del 27 novembre e 10 dicembre 2020 che hanno fatto registrare dei consumi molto più bassi e probabilmente anomali come già evidenziato in fase di analisi del dataset. Probabilmente questi due valori anomali falsificano in parte la predizione dato che i valori precedenti al 27 novembre sembrano essere predetti con molta più vicinanza al valore reale.
Le potenzialità di questo modello sembrano dunque più elevate rispetto ai precedenti e permette di effettuare anche predizioni più lunghe, ad esempio su più mesi o addirittura tutto il 2021, senza ottenere come risultato un valore costante come accadeva in precedenza. Si ritiene dunque che questo modello possa essere di utilità per la gestione del consumo elettrico in quanto, oltre a permettere un controllo a breve termine sull'andamento del consumo, permette di avere un quadro generale dei mesi a venire dando la possibilità di programmare eventuali lavori nei giorni in cui la rete è meno utilizzata minimizzando così i rischi di malfunzionamenti e disservizi per l'utenza finale. Inoltre sempre in quest'ottica a lungo termine può evidenziare in anticipo eventuali picchi di consumo permettendo così di mettere in atto politiche di prevenzione ed efficientamento.

6 Edificio U6: Analisi effettuate

In questa sezione verranno descritti i risultati ottenuti per l'edificio U6. I passaggi e la metodologia applicata è la medesima rispetto a quanto descritto per l'edificio U1, ci si concentrerà dunque maggiormente sulle differenze ottenute nei modelli.
Anche in questo caso si è iniziato studiando la struttura della serie storica attraverso la decomposizione spettrale, riportata in figura 8, in questo caso si nota come il trend vari nei diversi anni come osservato anche nell'analisi iniziale sui dati. Anche la stagionalità è diversa rispetto all'altro edificio, in questo caso infatti si nota una stagionalità forte, in negativo, per i mesi invernali.

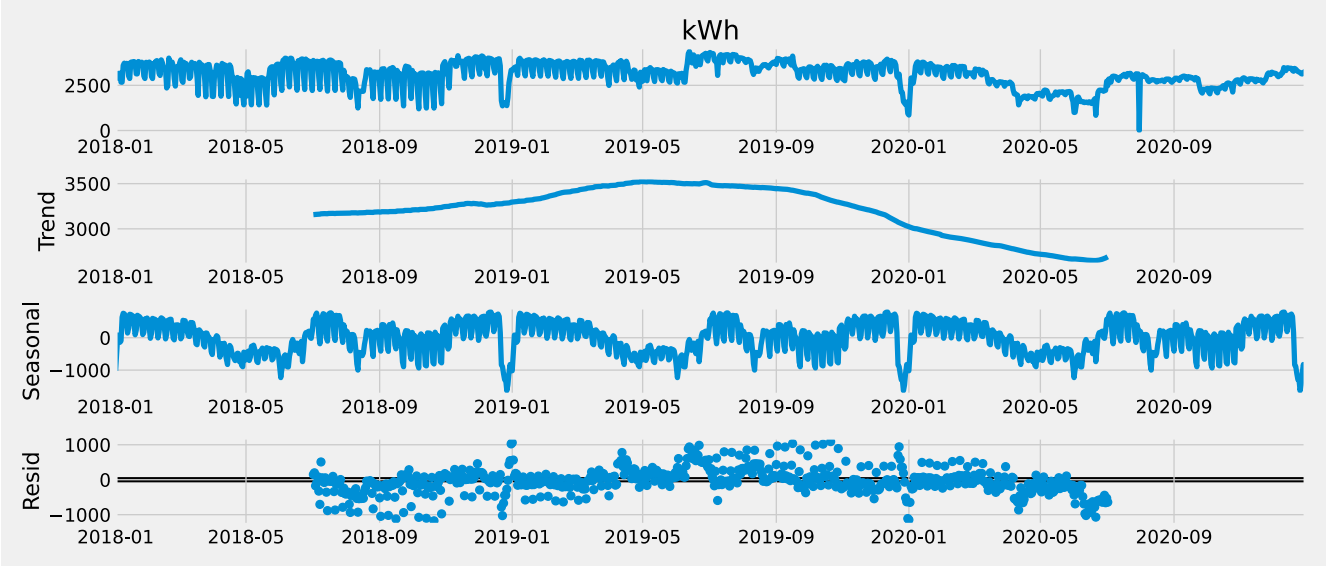

Figura 8: Decomposizione spettrale della serie storica per l'edificio U6

Si è poi testata la stazionarietà della serie con il test di Dickey-Fuller, e anche in questo caso è stato necessario differenziare la serie storica di un lag per ottenere la stazionarietà. Quindi anche per i modelli di U6 il parametro d sarà equivalente ad 1.
Per la scelta dei restanti parametri ci si è affidati nuovamente al metodo metodo `auto_arima` che permette di individuare i parametri migliori che minimizzino l'AIC. Per ridurre però lo spazio di ricerca ci si è basati sui grafici dell'autocorrelazione (ACF) e dell'autocorrelazione parziale (PACF) che hanno permesso di individuare i possibili valori candidati per i parametri che, al termine di questa fase, sono risultati essere nuovamente `p = 7, d = 1, q = 8`.

6.1 Rolling ARIMA (7,1,8) anni 2018-2019

La problematica sulla mancanza di anni per fittare correttamente modelli SARIMA è la medesima individuata per l'U1, quindi anche in questo caso si è utilizzato l'approccio "rolling" per la previsione. In particolare il codice utilizzato e i ragionamenti sono i medesimi di quanto esposto nel paragrafo 5.1.

6.2 Risultati Rolling ARIMA (7,1,8)

Il modello fittato risulta essere valido dal punto di vista statistico, infatti dai grafici di diagnostica, riportati in figura 9, risultano correlazioni all'interno della banda di normalità mentre l'adattamento alla normale non risulta ideale soprattutto nelle code, tuttavia la situazione è simmetrica tra le due code e quindi possiamo ritenere il modello abbastanza valido.

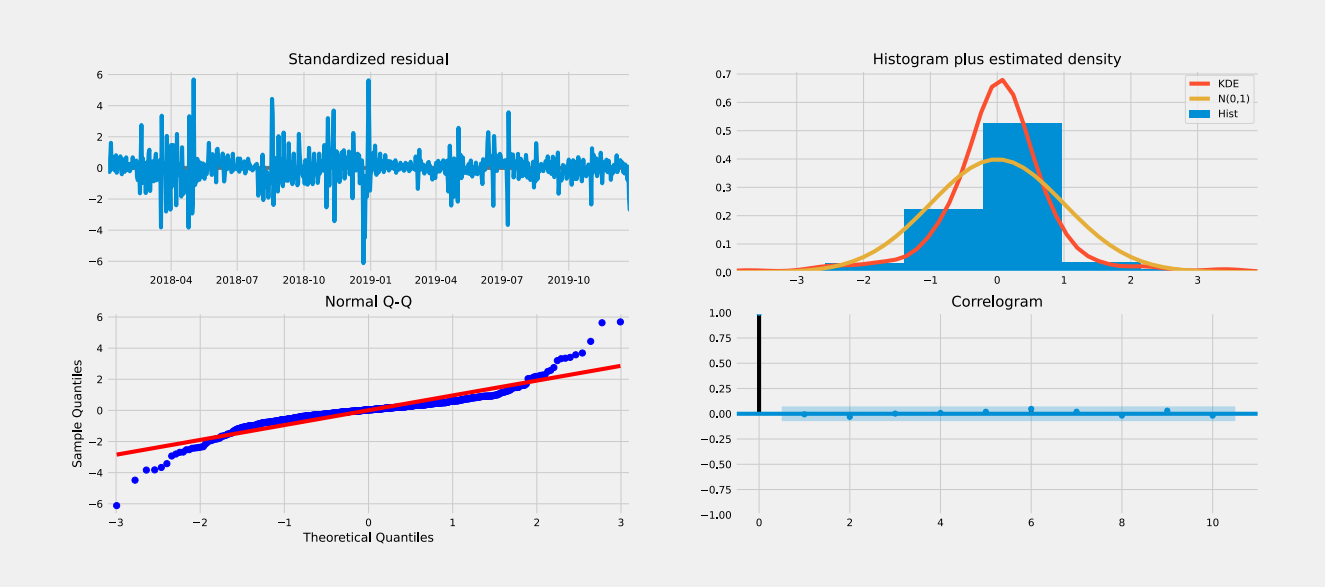

Figura 9: Rolling ARIMA (7,1,8) a 5 giorni: Grafici di diagnostica U6

Per quanto riguarda l'accuratezza delle previsioni, a differenza di quanto descritto per l'altro edificio in questo caso i risultati si sono rilevati migliori in quanto la stagionalità della serie è meno forte. Nello specifico l'indice R^2 calcolato tra i valori predetti dal modello e i valori realmente osservati si è rivelato essere molto alto e soddisfacente. Nel caso della finestra a cinque giorni si è ottenuto un R^2 pari a 0.63, un risultato buono che garantisce una previsione piuttosto accurata. In figura 10 è possibile confrontare graficamente l'andamento dei due valori, inoltre in figura è rappresentato l'intervallo di confidenza standard al 95%, si può notare come le predizioni siano molto corrette fino al periodo pre-natalizio mentre poi i valori predetti sono più alti rispetto a quelli realmente osservati.

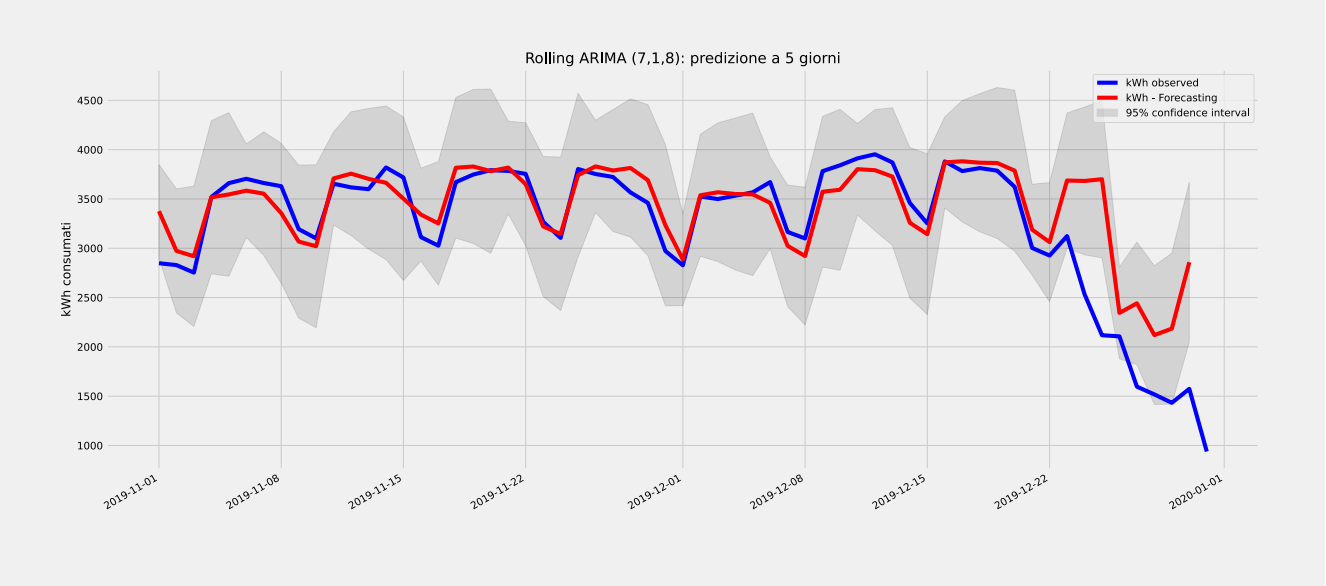

Figura 10: Rolling ARIMA (7,1,8) a 5 giorni: valori predetti fine 2019 U6

Si segnala inoltre che lo score risulta essere buono anche per finestre maggiori, ad esempio a 7 giorni, mentre per una finestra di un singolo giorno si ottengono risultati ottimi con uno score molto alto, vicino a 0.9.
Dato che i risultati ottenuti con finestra mobile erano molto buoni si è fatto un tentativo per prevedere l'andamento nel 2020 sfruttando come training iniziale gli anni 2018-2019 e aggiungendo step-by-step ulteriori valori spostando così la finestra. Questo tentativo è stato fatto considerando l'anno 2020 come anno realistico e non effetto da covid. Anche in questo caso l'approccio è risultato molto valido ottenendo degli score elevati fino ad una finestra a 15 giorni con la quale l'R^2 è risultato essere pari a 0.5, finestre più ridotte hanno fatto riscontrare score più elevati e molto incoraggianti per la previsione. Un riscontro completo sui valori ottenuti per l'indice R^2 in tutte le casistiche è consultabile nel appendice A, Tabella 2. Mentre in figura 11 si può visualizzare il grafico delle predizioni rispetto all'anno 2020 con una finestra fissata a 7 giorni.

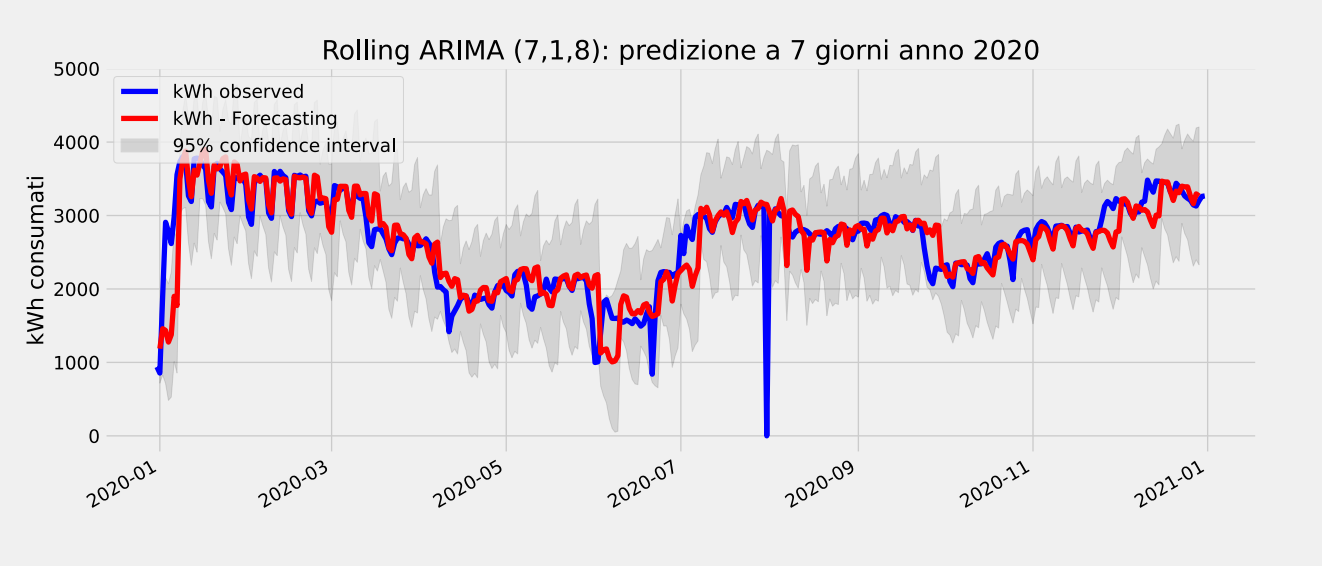

Figura 11: Rolling ARIMA (7,1,8) a 7 giorni: valori predetti per il 2020 U6

Si segnala inoltre che l'utilizzo della covariata "Weekday" in questo caso non migliora il modello, anzi si ottengono risultati peggiori nelle predizioni e quindi non è stato considerato.
Questi risultati sono dunque notevolmente migliori rispetto a quelli ottenuti per l'altro edificio e quindi si ritiene che possano davvero aiutare il personale nel controllo dei consumi e nel miglioramento degli stessi. Inoltre avendo una finestra più ampia a 15 giorni con score comunque buoni permettono anche di identificare periodi ideali per effettuare manutenzioni e/o lavori che richiedano delle piccole interruzioni nell'erogazione dell'energia minimizzando così l'impatto negativo per gli utenti finali.

6.3 SARIMA anni 2018-2020

Con l'introduzione nelle analisi dell'annata 2020 è stato possibile utilizzare i modelli SARIMA, ovvero includere nel modello anche la stagionalità della serie storica. Come in precedenza per l'implementazione di questi modelli si è utilizzato R che richiede meno tempo per il fitting dei modelli.
Anche in questo caso per la scelta dei parametri si è utilizzata la funzione `auto.arima` offerta dal pacchetto TSeries [7] di R, in questo caso i parametri sono risultati essere `order = (1,1,8), seasonal = (0, 1, 0, 365)` anche se con i parametri `order = (7,1,8), seasonal = (0, 1, 0, 365)` si è ottenuto un modello migliore in termini statistici ed è quindi stato preferito anche se l'AIC è leggermente maggiore.
Come per U1 si è proceduto fittando il modello con i valori da Gennaio 2018 a Ottobre 2020 mentre i restanti due mesi del 2020 sono stati riservati per la verifica dell'accuratezza delle predizioni.
Si rimanda all'appendice B, source code 4, per l'implementazione del codice in R.

6.4 Risultati SARIMA

Il modello SARIMA con `order = (7,1,8), seasonal = (0, 1, 0, 365)` ha richiesto circa 42 minuti di esecuzione per fittare tutti i valori del dataset in input, fino a ottobre 2020 in questo caso. Tuttavia il modello finale è risultato buono dal punto di vista statistico, come si può verificare in figura 12 anche se l'adattamento alla normale non è ottimale in cui si verifica il solito distanziamento soprattutto nelle code che però rimangono simmetriche.

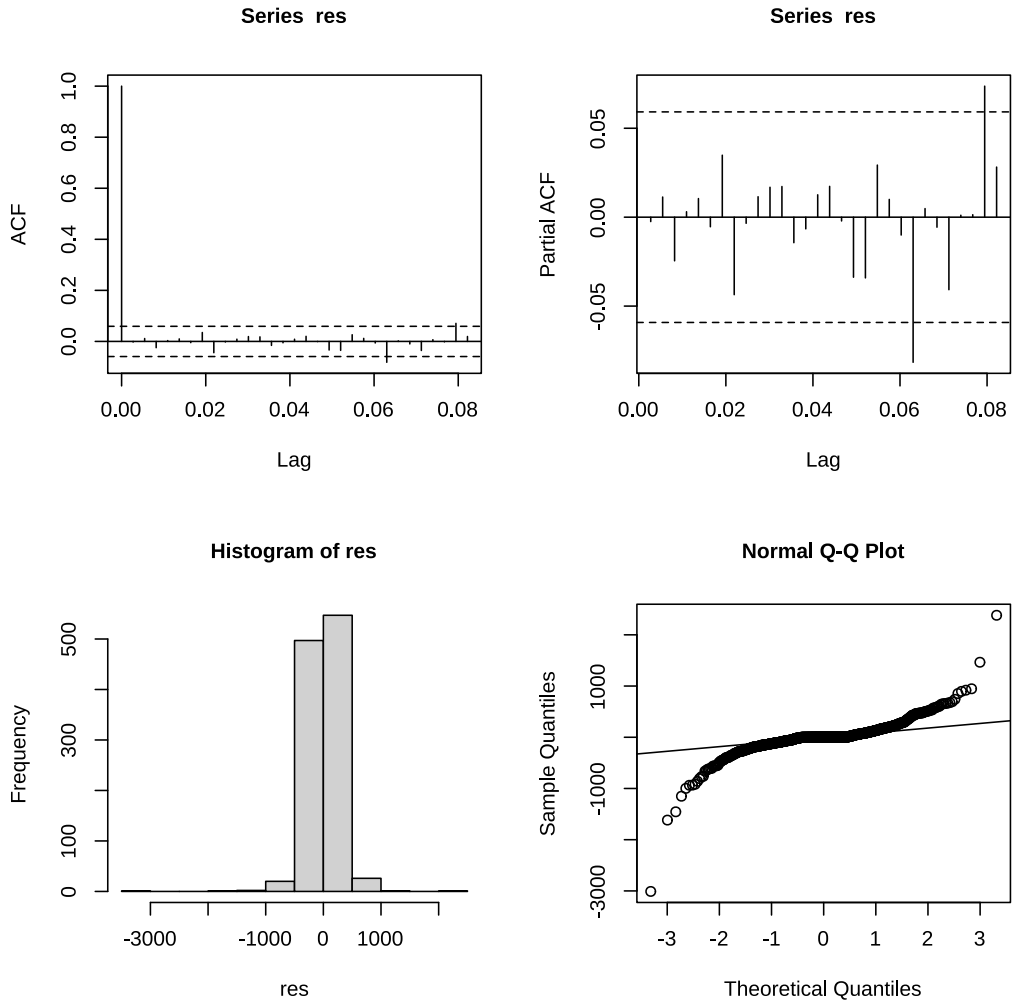

Figura 12: SARIMA (7,1,8) (0,1,0,365): Grafici di diagnostica per U6

Dal punto di vista della predizione tuttavia questi modelli sono risultati peggiori rispetto ai precedenti che applicavano il rolling con uno score R^2 molto basso pari a 0.06 negli ultimi due mesi del 2020. Il risultato notevolmente diverso in termini qualitativi, rispetto ad U1, fa pensare che in questo edificio l'impatto del Covid-19 sia stato notevolmente maggiore e quindi influenzi l'analisi. Sarebbe dunque interessante provare a riapplicare questo modello con annate diverse non influenzate dal covid.
Per le motivazioni appena descritte si ritiene che al momento questo modello non molto di utilità e non dia informazioni aggiuntive a chi raccoglie e gestisce i consumi energetici.

7 Ulteriori analisi: random forest e reti neurali

Visti i risultati dei modelli precedenti si è provato a seguire approcci neurali al fine di migliorare la finestra di predizione, si è deciso di testare anche l'algoritmo di machine learning Random Forest e la rete neurale ricorrente di tipo Long Short-Term Memory (LSTM). Nel primo caso, è stata implementata una versione di apprendimento supervisionato dell'algoritmo per effettuare la previsione basandosi solo sulla variabile del consumo giornaliero (kwh) sfruttando la libreria Python Scikit-learn [5] che offre dei metodi di apprendimento automatico. A tale proposito, trattandosi di una serie storica, si ha inizialmente optato per individuare un numero di lag ottimale per la variabile in esame. Tuttavia, tale passaggio è stato sostituito dalla tecnica di feature engineering della rolling window, decisamente più performante se assegnando un'ampiezza pari a 1 alla finestra. Inoltre, si è deciso di usare come test set il periodo corrispondente agli ultimi 5 mesi del 2020, quindi dal 1 agosto fino al 31 dicembre. In aggiunta, per quanto riguarda l'iperparametro relativo al numero di alberi, dopo alcune prove con diversi ordini di grandezza, si è inserito un valore pari a 100.
Successivamente, sempre tramite l'algoritmo delle random forest del pacchetto Scikit-learn, sono stati effettuati una serie di confronti inserendo diverse combinazioni di covariate (temperatura, vento, umidità) sempre abbinate a quella di output del consumo giornaliero (kWh), confrontate poi in termini di $MeanAbsoluteError$ nella previsione degli ultimi 5 mesi del 2020 ed in seguito interamente per tutto il 2020. Inoltre, si specifica che è stato mantenuto il valore del lag sempre pari a 1 e 1000 come numero di alberi, e che si ha anche usato il metodo della Walk Forward validation per cercare di migliorare ulteriormente l'attendibilità dei risultati considerando l'aspetto temporale dei dati. I dataset, inoltre, sono stai necessariamente trasformati in array grazie all'apposita funzione fornita dalla libreria NumPy [9] al fine di essere processati correttamente dalle altre funzioni utilizzate. Nell'appendice B, Source Code 3, è riportato il codice utilizzato per la realizzazione del algoritmo in Python.
Infine, come già riportato precedentemente, sono stati fatti dei tentativi implementando anche la rete neurale ricorrente LSTM, caratterizzata da un'apposita struttura che permette di cogliere eventuali dipendenze a lungo termine nei dati. Inizialmente è stato effettuato come step di pre-processing lo scaling delle variabili, affinché assumessero valore tra 0 e 1, grazie alla funzione "MinMaxScaler" presente in Scikit-learn. Banalmente si ha selezionato un lag pari a uno per tutte le variabili. Inoltre, è risultato vantaggioso mantenere sempre il numero di epoche uguale a 50, provando a variare, sulla base di quanto solitamente si evince dalla letteratura, il valore della batch size tra 8, 32 e 64 [1]. Per quanto concerne la Loss function, invece, è stata selezionato il $MeanAbsoluteValue$ essendo tendenzialmente robusto rispetto agli outlier, mentre come metodo di ottimizzazione quello dell'$AdaptiveMomentestimation$ (Adam), ovvero una variante della Discesa del Gradiente che ad ogni iterazione effettua una media mobile esponenziale sia dei gradienti che dei loro quadrati. Inoltre, per ogni versione della rete neurale è stato usato il $RootMeanSquaredError$ per individuare quella migliore, che quindi i valori più bassi indicano che i valori predetti si avvicinano maggiormente ai dati reali.

7.1 Risultati Random Forest e RNN LSTM

Relativamente ai risultati ottenuti dalla Random Forest per gli ultimi 5 mesi del 2020, è stato riportato uno score inaspettato pari ad un'$Accuracy$ di 0.999 per entrambi gli edifici. Per completezza e per curiosità è stata comunque effettuata una predizione usando l'intero 2020 come test set. Anche questa volta i risultati sono stati ottimi: uno score di 0.986 per l'edificio U1 e di 0.990 per l'U6. Osservando i grafici riportati in figura 13 e 14, è interessante notare, come i punti caratterizzati dai valori più difficoltosi da prevedere siano quelli in corrispondenza dei valori minimi. Infatti sembra che l'algoritmo ponga come valore minimo i picchi inferiori rilevati nel training set e che quindi non si immagina di riscontrare in futuro valori inferiori a questi. In sintesi, è anche opportuno notare come con questa strategia sia stato possibile ottenere performance notevoli nonostante ci si sia limitati a inserire e valutare un numero limitato di parametri. Tali risultati sono in linea con quanto prepostosi come obiettivo, inoltre essendo la predizione molto accurata e a lungo termine permettono ampi margini di manovra sia da un punto di vista attuativo, ad esempio cercando di ridurre il consumo energetico previsto qualora troppo elevato con politiche di risparmio energetico, sia dal punto di vista degli interventi in quanto permette con un buon anticipo di fissarli in giornate con meno richiesta energetica.

[1] Questi valori non sono casuali, solitamente infatti si scelgono potenze di 2 perché ideali per l'utilizzo di GPU per accelerare l'esecuzione del codice

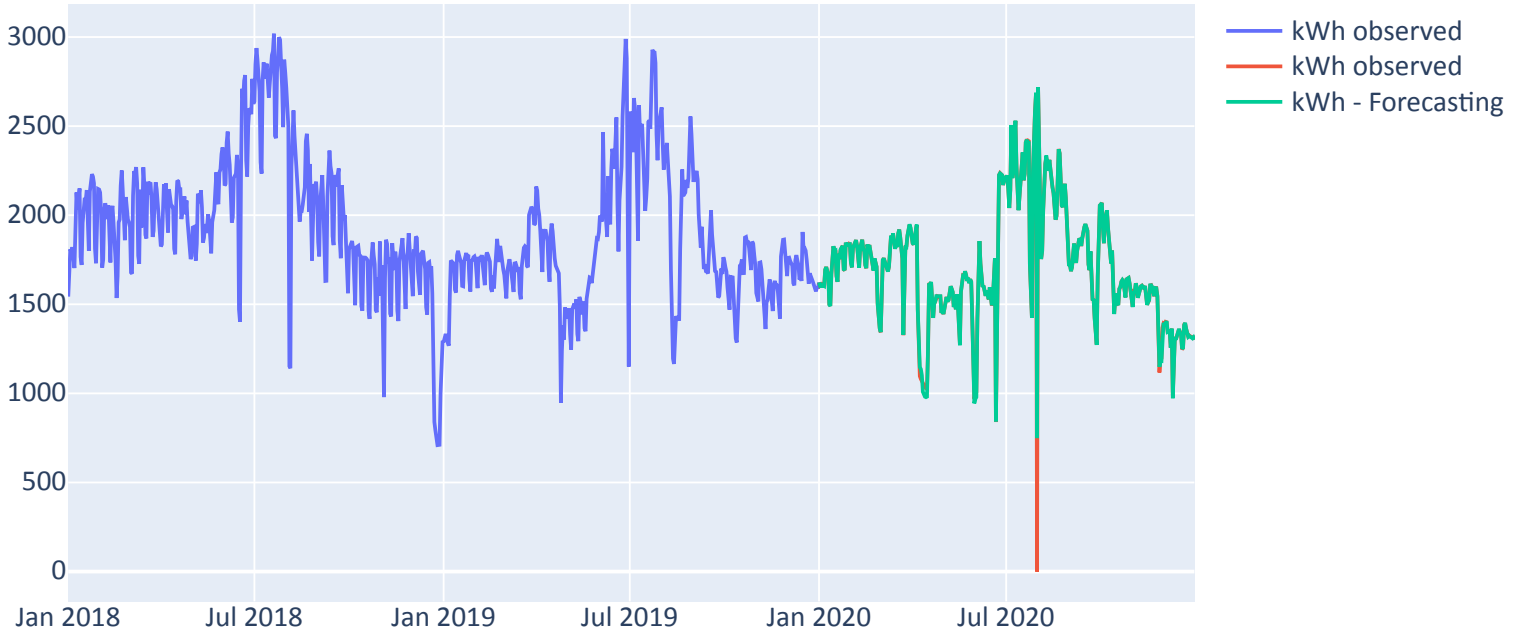

Figura 13: Random Forest: previsione del 2020 per U1

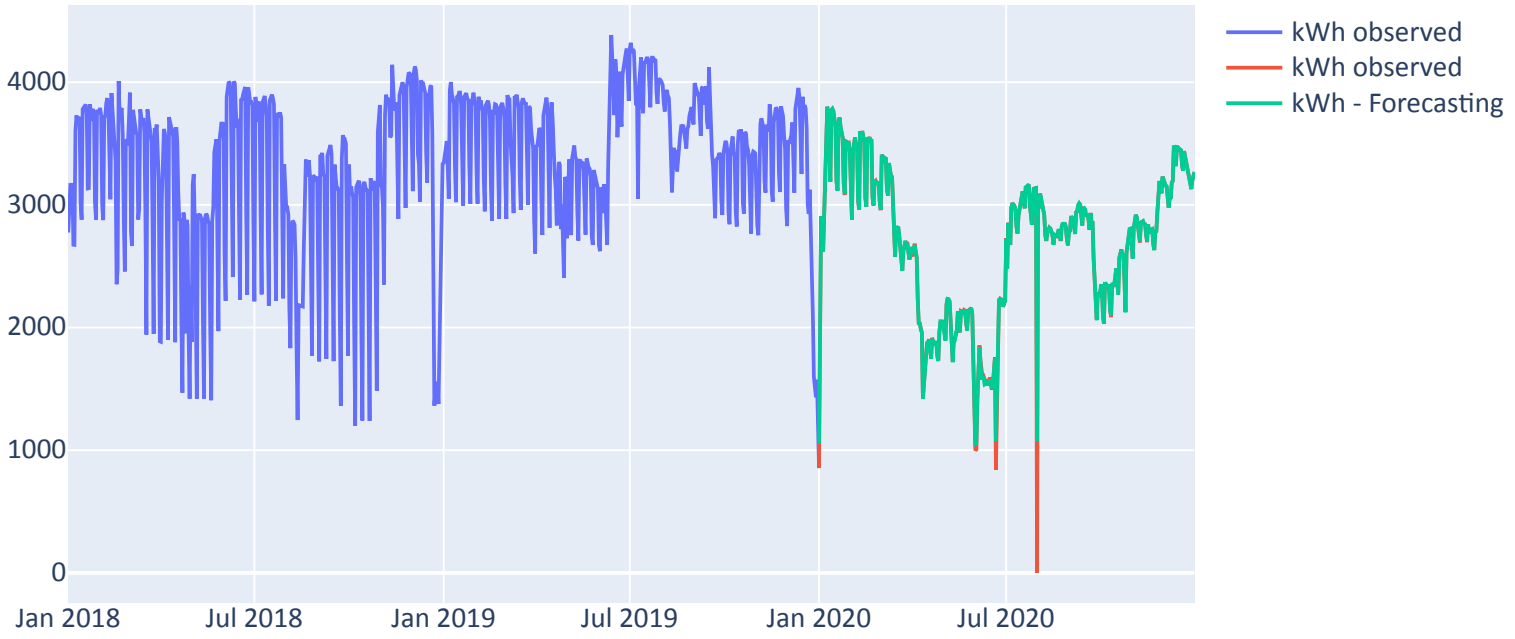

Figura 14: Random Forest: previsione del 2020 per U6

Per quanto concerne la versione multivariata di questo stesso algoritmo, ovvero inserendo molteplici covariate con diversi abbinamenti, invece, è emerso che è la variabile del vento che se associata ai kWh sembra comportare la riduzione più evidente del $MeanAbsoluteError$ per entrambi gli edifici (U1: 0.566; U6: 0.583); mentre inserendo, ad esempio, solo la temperatura abbinata ai kWh si otterrebbero rispettivamente dei valori pari a 6.561 e 7.282. Infine se si considerassero tutte (ovvero kWh, temperatura, vento, umidità) si otterrebbe una lieve riduzione rispetto al primo risultato proposto, cioè un MAE di 0.512 per l'U1 e di 0.525 per l'U6. Complessivamente, inoltre, i valori registrati per l'U1 sono sempre leggermente più bassi rispetto all'altro edificio. Si rimanda all'appendice A, tabella 5, per confrontare più nel dettaglio i risultati ottenuti con le varie configurazioni.
Esaminando i risultati ottenuti dalla rete neurale ricorrente, ci si vuole limitare solamente ad un rapido confronto tra le performance complessive dei due edifici. Per quanto riguarda l'U1, è emerso che la rete neurale più performante è quella che usa una batch size pari a 32, ottenendo un $RMSE$ di 104.117 sul test set, anche se con batch size pari solo a 8 risultava di 112.359. Considerando l'U6, invece, tutti i valori dell'$RMSE$ per i vari casi di batch size sono risultati più elevati rispetto a quelli registrati per l'altra struttura (come emerso anche con l'algortimo Random Forest), quello minimo, di 107.233, è stato ottenuto con batch size di 64. Nell'appendice A, tabella 6, sono riportati i diversi score ottenuti con le diverse dimensioni del batch size.

8 Conclusione e possibili sviluppi

Durante lo sviluppo dei modelli sono emerse diverse difficoltà dovute sia all'inesperienza nello specifico contesto implementativo delle serie storiche sia alla mancanza di alcune fonti di dati. Ad esempio gli anni a disposizione sono spesso risultati pochi rispetto ai tipi di algoritmi che si pensava di implementare forzando così alcune scelte, dall'altra parte la presenza della problematica Covid-19 ha introdotto problematiche nuove di più difficile comprensione che hanno richiesto uno studio più approfondito e una differenziazione degli algoritmi. In tal senso sarebbe stato utile avere più contesto specifico dei piani di efficientamento attuati da Bicocca per poter definire meglio i decrementi registrati nei diversi anni. Tuttavia nel complesso gli algoritmi sviluppati rispondono agli obiettivi preposti in fase iniziale e dunque possono effettivamente aiutare nella gestione e nel controllo dei consumi energetici. Nello specifico i modelli ARIMA e SARIMA offrono soluzioni più a breve termine con finestre temporali ridotte a 7/15 giorni, per periodi più lunghi i modelli non sembrano così performanti nella predizione.
Sono invece più confortanti e performanti i risultati ottenuti dai due algoritmi di machine learning, risulta indiscussa la qualità dell'algoritmo Random Forest, che nonostante non sia stato accompagnato da particolari step di pre-processing o di tuning degli iperparametri ha permesso di ottenere i risultati migliori in assoluto per le previsioni dei consumi energetici in entrambi gli edifici aumentando la finestra predittiva anche a tutto un anno garantendo così ai gestori molto più spazio di manovra per eventuali programmazioni sia di interventi alla rete sia di miglioramento. Mentre per i modelli LSTM, sebbene i risultati siano molto buoni, è importante notare che i dati integrati sono dati storici e quindi riportano effettivamente i valori meteo rilevati per quella giornata mentre nel caso di previsione di valori più avanti nel tempo si avrebbero a disposizione soltanto le previsioni meteo che, come ben risaputo, sono affidabili con una certa confidenza e inoltre sono disponibili per pochi giorni in avanti. Questo potrebbe ridurre la forza predittiva dei modelli LSTM peggiorandone l'accuratezza anche se in tal senso non è stato possibile effettuare alcun test perchè non è stato trovato alcun dataset storico delle previsioni meteo.
Per il futuro ci sarebbe la possibilità di migliorare questi modelli attraverso studi più approfonditi delle politiche di Bicocca e soprattutto testandoli su una quantità maggiore di dati, inoltre si potrebbe approfondire lo studio a livello micro, considerando, ad esempio, i periodi tipici che scandiscono la vita universitaria, sia a livello giornaliero che durante le festività o i periodi di lezioni/sessioni di esame. Tuttavia, data la moltitudine ed eterogeneità di corsi ed eventi che hanno luogo nei vari edifici dell'Ateneo, questa analisi forse non risulterebbe particolarmente agevole, anche semplicemente nel reperire tutte le informazioni.

Appendice A Tabelle

Score rolling arima U1 Si riporta di seguito una tabella riassuntiva esplicitante i valori dell'indice R^2 ottenuti con diverse ampiezze per la finestra di predizione per l'edificio U1. La tabella fa riferimento ai modelli spiegati nel paragrafo 5.1.

Modello	Step predizione	Training set	R^2 training set	Test set	R^2 test set
Rolling ARIMA (7,1,8)	5 giorni	Gen 2018 - Ott 2019	0.825	Nov-Dic 2019	0.515
Rolling ARIMA (7,1,8)	7 giorni	Gen 2018 - Ott 2019	0.822	Nov-Dic 2019	0.356
Rolling ARIMA (7,1,8) con covariata "weekday"	5 giorni	Gen 2018 - Ott 2019	0.843	Nov-Dic 2019	0.488
Rolling ARIMA (7,1,8) con covariata "weekday"	7 giorni	Gen 2018 - Ott 2019	0.843	Nov-Dic 2019	0.392
Rolling ARIMA (7,1,8)	5 giorni	Gen 2018 - Dic 2019	0.762	Gen-Dic 2020	0.476
Rolling ARIMA (7,1,8)	7 giorni	Gen 2018 - Dic 2019	0.761	Gen-Dic 2020	0.430
Rolling ARIMA (7,1,8)	15 giorni	Gen 2018 - Dic 2019	0.756	Gen-Dic 2020	0.310
Rolling ARIMA (7,1,8) con covariata "weekday"	7 giorni	Gen 2018 - Dic 2019	0.761	Gen-Dic 2020	0.433

Tabella 1: Risultati Rolling ARIMA (7,1,8) per l'edificio U1

Score rolling arima U6 Si riporta di seguito la medesima tabella riassuntiva esplicitante i valori dell'indice R^2 ottenuti con diverse ampiezze per la finestra di predizione per l'edificio U6. La tabella fa riferimento ai modelli spiegati nel paragrafo 6.1.

Modello	Step predizione	Training set	R^2 training set	Test set	R^2 test set
Rolling ARIMA (7,1,8)	1 giorno	Gen 2018 - Ott 2019	0.839	Nov-Dic 2019	0.887
Rolling ARIMA (7,1,8)	5 giorni	Gen 2018 - Ott 2019	0.833	Nov-Dic 2019	0.630
Rolling ARIMA (7,1,8)	7 giorni	Gen 2018 - Ott 2019	0.834	Nov-Dic 2019	0.528
Rolling ARIMA (7,1,8) con covariata "weekday"	7 giorni	Gen 2018 - Ott 2019	0.830	Nov-Dic 2019	0.214
Rolling ARIMA (7,1,8)	5 giorni	Gen 2018 - Dic 2019	0.859	Gen-Dic 2020	0.732
Rolling ARIMA (7,1,8)	7 giorni	Gen 2018 - Dic 2019	0.859	Gen-Dic 2020	0.688
Rolling ARIMA (7,1,8)	15 giorni	Gen 2018 - Dic 2019	0.859	Gen-Dic 2020	0.5

Tabella 2: Risultati ARIMA (7,1,8) con e senza Rolling per l'edifico U6

Correlazione di Pearson tra variabili integrate e rilevazioni U1 Si riporta di seguito la tabella con i valori della correlazione di pearson ottenuta tra le variabili integrate, riportanti temperatura, vento e umidità, e i valori del consumo elettrico registrati.

Variabili	kWh	Temperatura	Umidità	Vento
kWh	1.000	0.518	-0.296	0.146
temperatura	0.518	1.000	-0.438	0.228
umidità	-0.296	-0.438	1.000	-0.178
vento	0.146	0.228	-0.178	1.000

Tabella 3: Confronto tra variabili integrative e kWh consumati per l'edificio U1

Correlazione di Pearson tra variabili integrate e rilevazioni U6 Si riporta di seguito la medesima tabella con i valori della correlazione di pearson per l'edificio U6.

Variabili	kWh	Temperatura	Umidità	Vento
kWh	1.000	-0.121	0.021	-0.036
temperatura	-0.121	1.000	-0.438	0.228
umidità	0.021	-0.438	1.000	-0.178
vento	-0.036	0.228	-0.178	1.000

Tabella 4: Confronto tra variabili integrative e kWh consumati per l'edificio U6

Confronto risultati Random Forest La seguente tabella riporta il confronto degli score, in termini di MAE, ottenuti per l'edificio U1 ed U6 in base alle variabili considerate.

Variabili	MAE U1	MAE U6
kWh + temperatura + umidità + vento	0.512	0.525
kWh + temperatura + vento	0.540	0.540
kWh + temperatura	6.561	7.282
kWh + vento	0.566	0.583
kWh + temperatura + umidità	11.453	13.197
kWh + vento + umidità	0.509	0.535

Tabella 5: Confronto score random forest tra gli edifici

Contronto RNN LSTM La tabella sottostante riporta gli score, in termini di $RMSE$, ottenuti per l'edificio U1 ed U6 in base alle dimensione del batch size considerata.

Batch size	RMSE U1	RMSE U6
8	112.359	282.990
32	104.117	126.041
64	105.541	107.233

Tabella 6: Confronto score LSTM tra gli edifici

Appendice B Codice Python

Rolling ARIMA(7,1,8) Si riporta di seguito il codice Python utilizzato per predire i valori con una finestra mobile di n giorni, come descritto nei paragrafi 5.1 e 6.1.

Source Code 1: Rolling ARIMA (7,1,8)

```
# salvo le predizioni fatte step by step
prediction_df_5 = pd.DataFrame()
# Grandezza finestra di previsione
num_steps_pred = 5

# continuo a ri-fittare il modello finchè non arrivo a predire gli ultimi 5 gg del 2019
for i in range(0, round(max_iters)):
  # In questo caso SARIMAX = ARIMA perchè non è specificata nessuna stagionalità!
  ar = SARIMAX(df.kWh[:end_index], order=(7,1,8)).fit()

  # prevedo a 5 gg avanti con confidenza 95%
  end_pred = end_index + num_steps_pred -1
  temp = ar.get_prediction(start = end_index, end = end_pred).summary_frame()
  # salvo le predizioni effettuate man mano
  prediction_df_5 = prediction_df_5.append(temp)

  # proseguo con la previsione di 5gg
  end_index += num_steps_pred
```

Rolling ARIMA(7,1,8) con covariata Il seguente codice riporta invece l'implementazione della variante citata nel paragrafo 5.1, nella quale viene utilizzata anche la covariata "Weekday" per predire l'andamento dei consumi.

Source Code 2: Rolling ARIMA (7,1,8) con covariata "Weekday"

```
# salvo le predizioni fatte step by step
prediction_df_cov = pd.DataFrame()
# Grandezza finestra di previsione
num_steps_pred = 7
# salvo la lista contenente i valori di covariata
exog_val = df_U1['Weekday'].values

# continuo a ri-fittare il modello finchè non arrivo a predire gli ultimi 7 gg del 2019
for i in range(0, round(max_iters)):
  # fitto il modello passando anche i valori della covariata
  ar = SARIMAX(df_U1_19.kWh[:end_index], order=(7,1,8), exog=exog_val[:end_index]).fit()

  # prevedo a 7 gg con confidenza 95%
  end_pred = end_index + num_steps_pred -1

  # inserisco il try perchè i giorni non sono divisibili in modo intero per 7
  # in questo modo gli ultimi valori che rimangono fuori e darebbero errore
  # nella predizione vengono saltati.
  # L'errore è dovuto alla mancanza dei valori per la covariata, in quanto si
  # fermano al 2019-12-31 mentre la predizione finale 'sfora' anche ai primi
  # giorni del 2020
```

```
  try:
    temp = ar.get_prediction(start = end_index, end = end_pred,
                             exog=exog_val[end_index:end_pred+1].reshape(7,1)).summary_frame()
    prediction_df_cov = prediction_df_cov.append(temp)
  except:
    pass

  # proseguo con la previsione di 7gg con covariata
  end_index += num_steps_pred
```

Random Forest per serie storica univariata Nel codice riportato in questo paragrafo si può vedere l'implementazione dell'algoritmo Random Forest effettuando la previsione dei consumi energetici (kWh) senza l'ausilio di altre covariate.

Source Code 3: Random Forest per previsione variabile "kWh"

```
# Impostazione della finestra di ampiezza 1 per effettuare la previsione
# sulla colonna del dataset riferita ai consumi, eliminando eventuali valori nulli
df_forecasting["var_rolling"] = df_forecasting[kWh].rolling(window = 1).mean()
df_forecasting= df_forecasting.dropna()
#Assegnare la colonna dei consummi e la sua versione
#ritardata rispettivamente alle variabili y e x
x=df_forecasting.iloc[:,1:]
y=df_forecasting.iloc[:,0]
# Suddivisione del dataset tra train e test sulla base del valore
# della soglia precedentemente definita come una data che compare nel dataset
# (nel primo tentativo corrisponde all' 1/08/2020 e nel secondo al 1/01/2020)
x_train, x_test = x.loc[x.index < soglia], x.loc[x.index >= soglia]
y_train, y_test = y.loc[y.index < soglia], y.loc[y.index >= soglia]
#Fitting del Random Forest con 100 alberi
mdl = RandomForestRegressor(n_estimators=100)
mdl.fit(x_train, y_train)
pred=mdl.predict(x_valid)
predict[kWh+"_valid"]=y_valid.values
predict[kWh+"_predict"]=pred
# Preparare il valore dello score sui dati di test
s=mdl.score(x_test, y_test)
score.append([kWh, s])
score
```

Appendice C Codice R

SARIMA (7,1,8)(0,1,0,365) Si riporta di seguito il codice R utilizzato per predire i valori utilizzando modelli SARIMA, come descritto nei paragrafi 5.3 e 6.3.

Source Code 4: SARIMA (7,1,8)(0,1,0,365)

```
# leggo il dataset
df_U1_R <- read.csv("dataset_U6.csv", sep=",")
# estraggo la colonna contenente i kWh
kWh <- ts(df_U1_R[, 'kWh'], frequency=365)

# fitto il modello tenendo due mesi finali per il controllo
ar <- arima(kWh[0:1035], order=c(7,1,8),
              seasonal= list(order = c(0, 1, 0), period = 365))

# visualizzo i dati sul modello
print(summary(ar))

# effettuo la predizione sui valori mancanti del 2020
r_forecast_2020 <- forecast(ar, h=61)
# visualizzo il grafico
plot(r_forecast_2020, main="Seasonal ARIMA(5,1,2)(0,1,0,365): Predizioni 2020",
     ylab="kWh consumati", xlab = "Date")

# Visualizzo i grafici di diagnostica del modello
res <- ar$res
par(mfrow=c(2,2),cex.axis=1, cex.lab=1, cex.main=1, fg="black",
col.axis="black", col.lab="black", col.main="black", bg="white")
acf(res, ci.col="black")
pacf(res, ci.col="black")
hist(res)
qqnorm(res)
qqline(res)
shapiro.test(res)
```

9 Bibliografia

[1] Università degli Studi di Milano-Bicocca. *Energy management*. 2021. URL: https://www.unimib.it/ateneo/energy-management (visitato il 10/06/2021).

[2] Python Core Team. *Python: A dynamic, open source programming language*. Python Software Foundation. 2015. URL: https://www.python.org/..

[3] Taylor G. Smith et al. *pmdarima: ARIMA estimators for Python*. [Online; accessed 07-06-2021]. 2017. URL: http://www.alkaline-ml.com/pmdarima.

[4] Skipper Seabold e Josef Perktold. "statsmodels: Econometric and statistical modeling with python". In: *9th Python in Science Conference*. 2010.

[5] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[6] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2020. URL: https://www.R-project.org/.

[7] Adrian Trapletti e Kurt Hornik. *tseries: Time Series Analysis and Computational Finance*. R package version 0.10-48. 2020. URL: https://CRAN.R-project.org/package=tseries.

[8] Rob Hyndman et al. *forecast: Forecasting functions for time series and linear models*. R package version 8.15. 2021. URL: https://pkg.robjhyndman.com/forecast/.

[9] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (set. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[10] Aditi Mittal. *Medium:Understanding RNN and LSTM*. 2019. URL: https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e.

[11] Hafidz Zulkifli. *towardsdatascience:Multivariate Time Series Forecasting Using Random Forest*.

[12] Benjamin Goehry et al. "Random Forests for Time Series". working paper or preprint. Feb. 2021. URL: https://hal.archives-ouvertes.fr/hal-03129751.

[13] Hyndman, R.J., and Athanasopoulos, G. *Forecasting: principles and practice, 2nd edition*. 2018. URL: oTexts.com/fpp2 (visitato il 10/06/2021).