Università degli studi di Milano-Bicocca

Corso di Laurea Magistrale in Data Science, Anno Accademico 2022-2023

Report

Streaming Data Management and Time Series Analysis Project

Author:

Emanuela Elli - 892901 - e.elli@campus.unimib.it

Abstract

Lo scopo di questo progetto è analizzare la serie storica relativa alle misurazioni del consumo di elettricità di una città del Marocco per prevederne l'andamento. Il seguente lavoro è stato effettuato tramite l'utilizzo di modelli lineari, quali ARIMA e UCM, e modelli non lineari, tra cui l'algoritmo di classificazione/regressione KNN e reti neurali ricorrenti LSTM. Sono stati utilizzati, inoltre, due linguaggi per la creazione e l'analisi di questi modelli, in particolare il linguaggio R per lo sviluppo di ARIMA, UCM e KNN ed il linguaggio Python, in particolare nell'ambiente Google Colab, per LSTM. I risultati ottenuti dall'esecuzione di tali modelli sono stati poi confrontati tramite il Mean Absolute Error (MAE) e il Mean Absolute Percentage Error (MAPE). Questi valori sono serviti sia per confrontare il comportamento tra le diverse famiglie, ma anche per valutare la bontà dei parametri scelti all'interno della stessa famiglia di modelli.

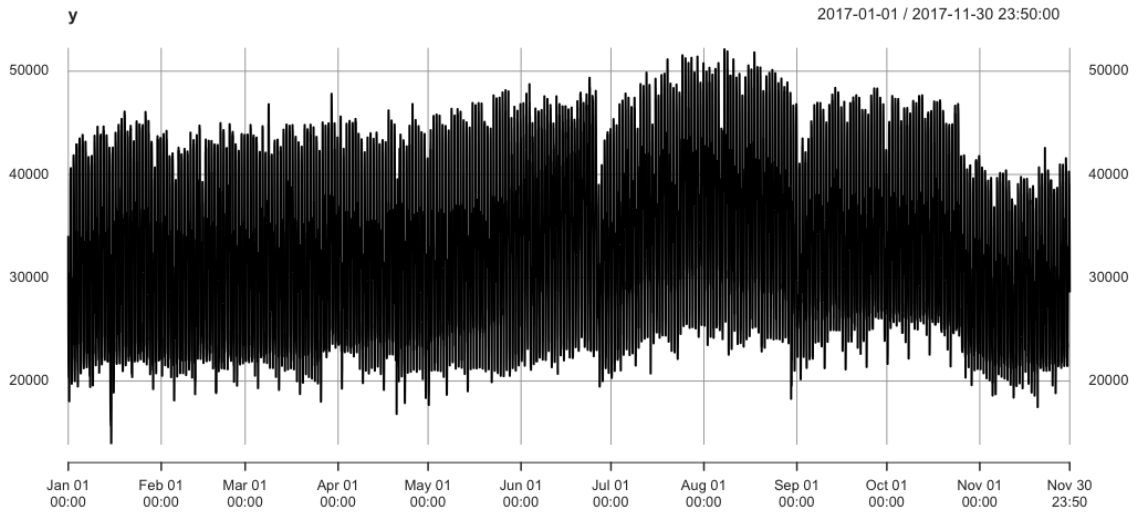
Pre-processing e analisi esplorativa

Per lo svolgimento di questo lavoro è stato fornito un dataset relativo alle misurazioni di consumo di elettricità di una città del Marocco (la conoscenza del luogo geografico è necessaria affinchè si possa valutare l'effetto dei giorni festivi per la cultura considerata). All'interno di tale set di dati sono presenti 48096 osservazioni e due variabili:

- `date`, ovvero la stringa codificante la data-ora della misurazione, in formato "`dd/mm/yyyy HH:MM:SS`";
- `power`, ovvero il consumo energetico rilevato.

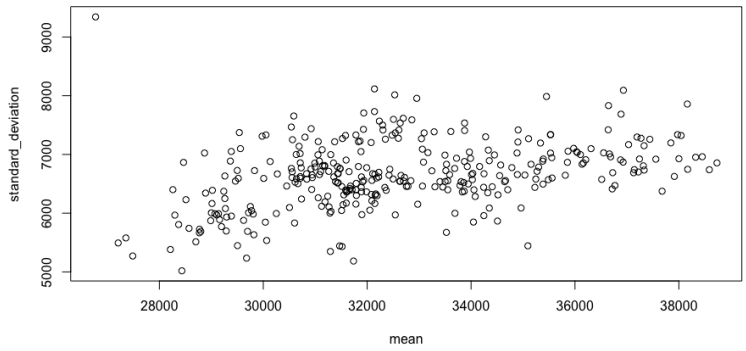
Tali misurazioni coprono il periodo dal "`01/01/2017 00:00:00`" fino al "`30/11/2017 23:50:00`", con osservazioni ogni 10 minuti. Il mese mancante all'interno della serie storica, ovvero Dicembre, è oggetto della previsione richiesta.

Come prima analisi si è verificato che le osservazioni non presentassero valori nulli. Confermata l'assenza di valori nulli, si è proceduto alla creazione di un oggetto *time series* e alla prima visualizzazione dei dati grezzi:



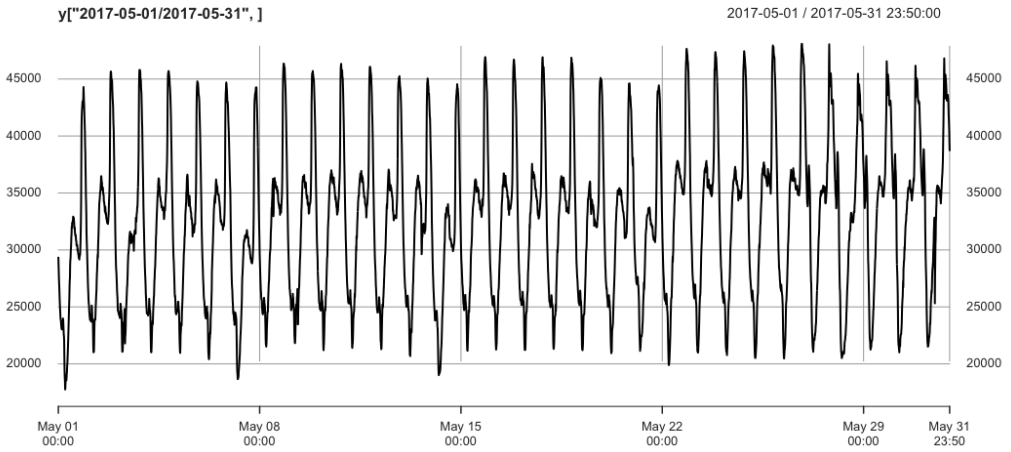
Da una prima osservazione si potrebbe affermare che è presente un leggero trend crescente nei mesi estivi e decrescente da Agosto a Novembre.

Per verificare la stazionarietà in varianza della serie storica si controlla il grafico di Box-Cox per osservare se è presente una relazione tra la media e la deviazione standard.



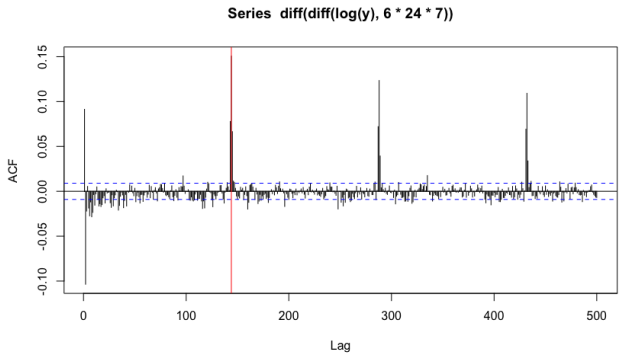
Dall'immagine sembra esserci una leggera relazione lineare crescente.

Per quanto riguarda, invece, la stazionarietà in media si osserva, dal seguente grafico in cui si mostra solamente un periodo della serie storica (in particolare il mese di Maggio), una stagionalità giornaliera e settimanale (oltre a quella mensile).

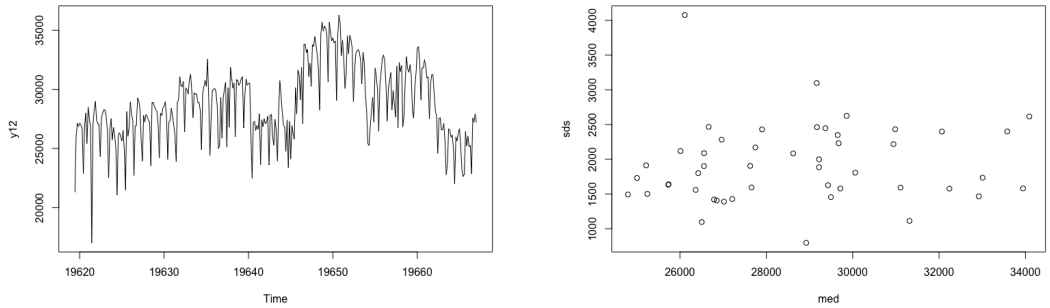


Date le precedenti osservazioni, si è provato ad applicare la trasformazione del logaritmo per correggere il trend linearmente crescente e sono state applicate due differenze, una stagionale e una semplice, in modo da ottenere la stazionarietà sia in media che in varianza della serie storica.

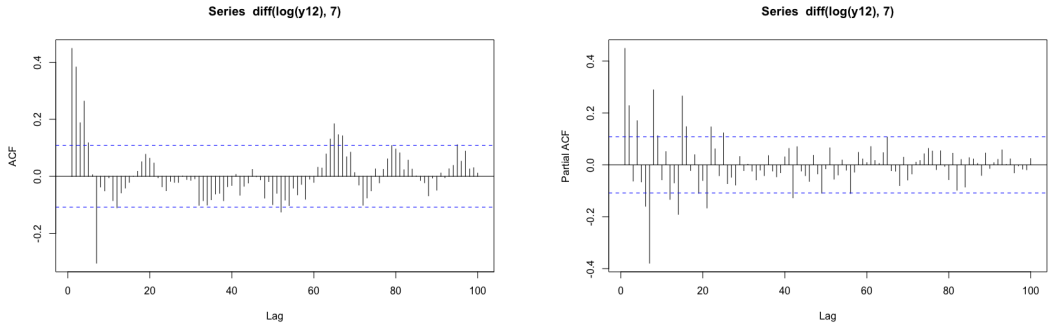
Nonostante le trasformazioni e la verifica tramite i test di Dickey-Fuller e KPSS, nell'ACF risultano esserci ancora dei picchi molto forti ogni 144 lag circa.



Siccome la serie storica a disposizione presenta più stagionalità (ovvero giornaliera, settimanale e mensile), si è deciso di considerare un orario specifico per ogni giorno (in particolare le ore "`09:00:00`") in maniera tale da eliminare la stagionalità giornaliera e di avere delle esecuzioni dei modelli più veloci.



In questo caso si è dovuta eliminare la non stazionarietà in media e la stagionalità presente, mentre per quanto riguarda la non stazionarietà in varianza si è osservato un leggero trend lineare crescente. Per questo motivo si è deciso di applicare il logaritmo e la differenza stagionale per rendere la serie storica stazionaria ottenendo i grafici della ACF e PACF mostrati di seguito.



Il grafico della ACF presenta un andamento decrescente verso 0, pertanto è sicuramente possibile inserire nel modello un AR e tramite il grafico della PACF si può ipotizzare sia meglio la scelta di un AR(2). Anche la PACF risulta avere un andamento verso 0 pertanto si può ipotizzare di inserire un MA all'interno del modello.

Modelli Arima

Per lo sviluppo di modelli ARIMA si è scelto l'approccio di identificare il modello più adatto per i dati e replicarlo per tutti gli orari del giorno, creando così 144 modelli (6*24) per ottenere le previsioni orarie giornaliere. Inoltre sono state testate diverse combinazioni di modelli ARIMA, considerando fasce orarie differenti, in modo da scegliere i parametri più ottimali.

Per ottenere delle previsioni migliori, sono state create delle variabili dummies per rappresentare i giorni festivi o particolari outliers presenti nella serie storica (ad esempio si nota uno shock importante in corrispondenza del 26 - 27 giugno, anch'esso considerato con una variabile dummy). Di seguito vengono elencate le festività considerate nell'analisi.

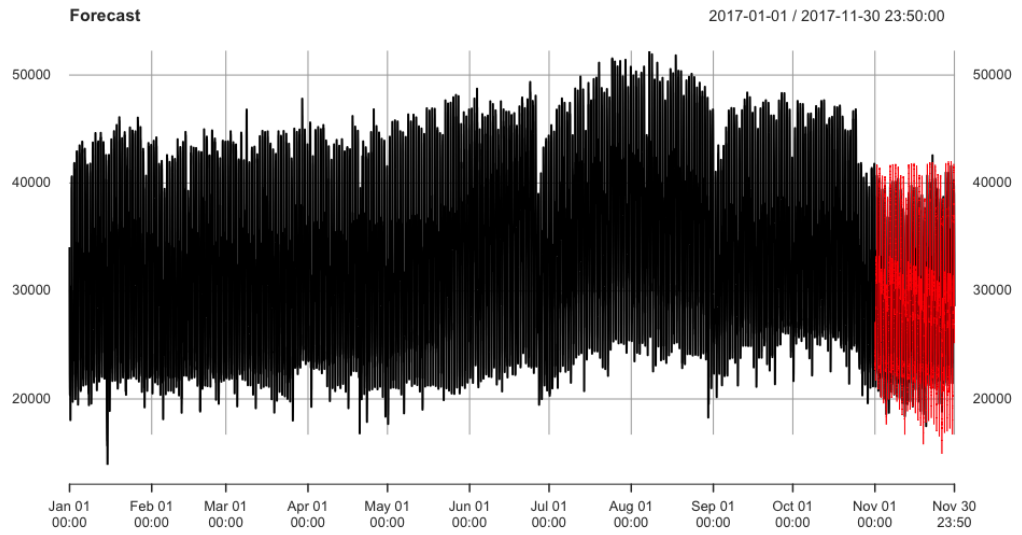
LIST OF HOLIDAYS IN MOROCCO IN 2017

Day	Date	Holiday Name	Type	Comments
Sunday	Jan 01	New Year's Day	National Holiday	
Wednesday	Jan 11	Proclamation of Independence	National Holiday	Anniversary of the Independence Manifesto of 1944
Monday	May 01	Labour Day	National Holiday	International Workers' Day
Monday	Jun 26	Eid Sghir	National Holiday	End of Ramadan
Tuesday	Jun 27	Eid Sghir Holiday	Government Holiday	Government only
Sunday	Jul 30	Enthronement	National Holiday	King Mohammed VI's coronation in 1999
Monday	Aug 14	Oued Ed-Dahab Day	National Holiday	Allegiance Day
Sunday	Aug 20	Revolution Day	National Holiday	Marks King Mohammed V's exile in 1953
Monday	Aug 21	Youth Day	National Holiday	
Friday	Sep 01	Eid Kbir	National Holiday	Feast of the Sacrifice
Saturday	Sep 02	Eid Kbir Holiday	Government Holiday	Banks and Government only
Thursday	Sep 21	Fatih Muharram	National Holiday	Islamic New Year
Monday	Nov 06	Green March Day	National Holiday	
Saturday	Nov 18	Independence Day	National Holiday	National Day
Friday	Dec 01	Eid Al Mawled	National Holiday	Birthday of Prophet Muhammad

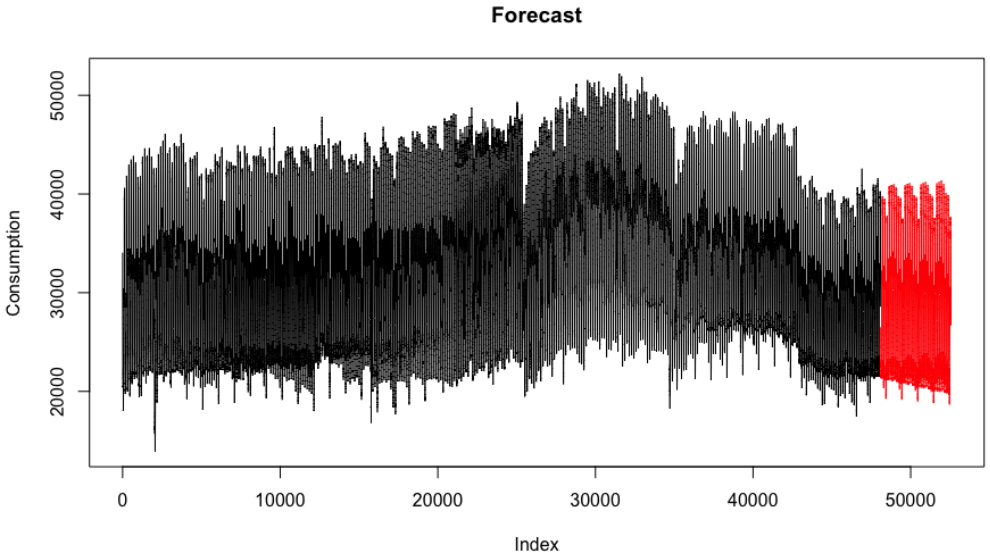
Nella tabella vengono riportati alcuni modelli testati:

Modello	**MAE**	**MAPE**	**REG**
ARIMA(1, 0, 1)(2, 1, 0)$_7$	1363.44	4.91	dummy
ARIMA(1, 0, 1)(2, 1, 0)$_7$ + drift	1321.74	4.79	dummy
ARIMA(2, 1, 0)(0, 1, 1)$_7$	1031.17	3.65	dummy

Il modello scelto per effettuare le previsioni per il mese di dicembre è quindi ARIMA(2, 1, 0)(0, 1, 1)$_7$. Il grafico sottostante mostra in nero i dati osservati e in rosso le previsioni effettuate, per il mese di Novembre, dal modello selezionato.



Nel grafico sottostante, invece, vengono mostrate le previsioni richieste per il mese di Dicembre.



Modelli UCM

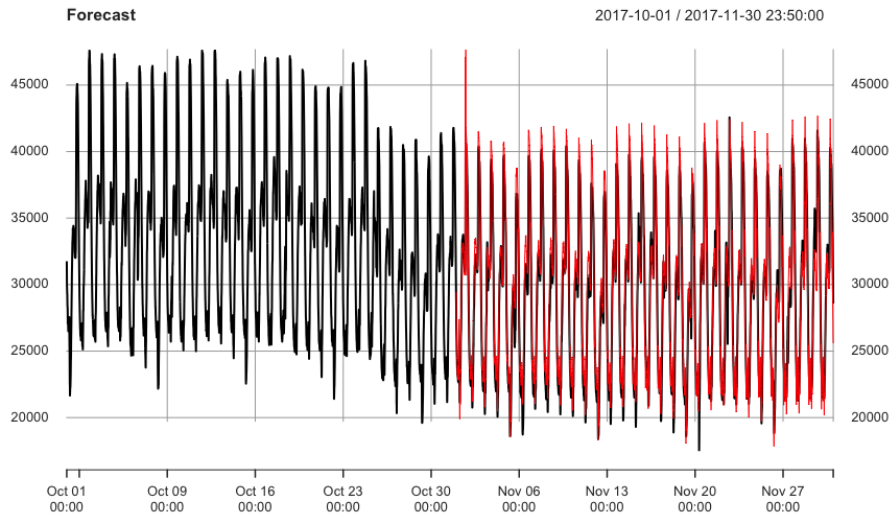
Il medesimo approccio, utilizzato per i modelli ARIMA, è stato adottato anche per i modelli UCM. Pertanto, una volta identificati i modelli candidati, questi sono stati utilizzati per tutti gli orari.

Partendo dalle conoscenze acquisite durante la fase di costruzione dei modelli ARIMA, si è deciso di utilizzare modelli con una componente stagionale a dummy stocastiche (per modellare la stagionalità settimanale, ovvero s=7) ma anche una seconda stagionalità (annuale, s=365) modellata con componenti trigonometriche. Sono state inoltre utilizzate variabili dummies, precedentemente citate, per comprendere le festività e gli outliers.

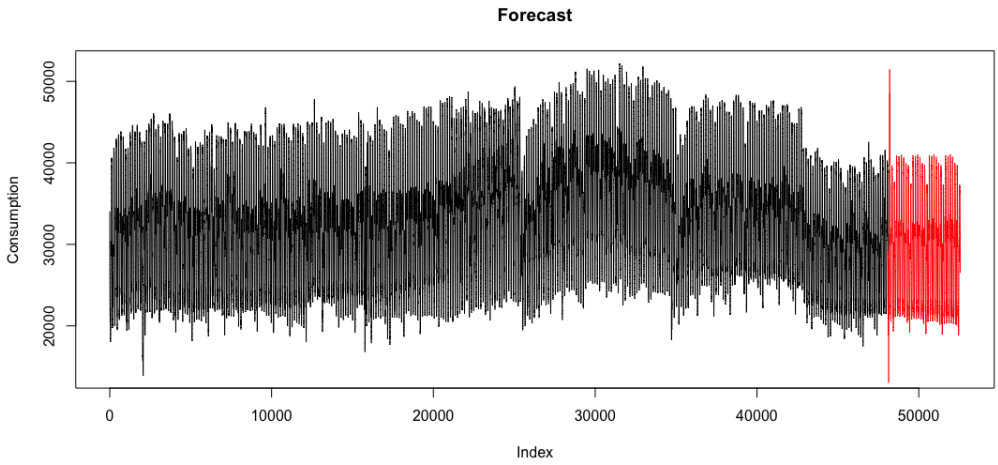
Di seguito è possibile osservare i risultati, in termini di MAE e MAPE, di alcune prove effettuate:

Modello	**MAE**	**MAPE**
Dummies + LLT + seasonal 7 dummy	1253.39	4.49
Dummies + LLT + seasonal 7 dummy + seasonal 365 trigonometric 8 harmonics	1198.71	4.28
Dummies + LLT + seasonal 7 dummy + seasonal 365 trigonometric 2 harmonics	1080.84	3.83

Il migliore risulta essere il terzo ovvero il modello che comprende entrambe le stagionalità (annuale e settimanale) impostando il numero di sinusoidi a 2. Le previsioni effettuate con tale modello sono mostrate nel grafico sottostante (per esaminare meglio la previsione viene mostrato solo un sotto insieme dei dati di training ed in cui vengono evidenziate in rosso le previsioni).



Nel grafico sottostante, invece, vengono mostrate le previsioni richieste per il mese di Dicembre.



Modelli ML

Per i modelli di machine learning, che appartengono invece alla famiglia di modelli non lineari, si è deciso di testare i seguenti approcci:

- KNN;
- LSTM.

In questo caso non è stato seguito l'approccio adottato precedentemente per i modelli lineari, ma è stato utilizzato l'intero set di dati per effettuare le previsioni. È importante evidenziare, inoltre, che all'interno dei modelli non lineari non è possibile dichiarare la presenza di eventi specifici (ad esempio festività e outliers), come si effettuava invece nei modelli lineari. In questo modo i modelli ARIMA e UCM sono in grado di analizzare più fedelmente eventi imprevisti, mentre nei modelli non lineari viene effettuato in maniera autonoma ma necessitano di una quantità di dati molto superiore per far sì che venga rilevato un particolare comportamento.

KNN

L'algoritmo K-Nearest Neighbors viene utilizzato sia per task di classificazione che per quelli di regressione, per cui è adatto anche per le serie storiche. Tale modello si basa sulla semplice regola di classificazione che consiste nell'incorporare nuovi campioni nello spazio di addestramento prima di classificarli sulla base della maggioranza. Il risultato finale fa sì che i punti dati simili siano vicini in prossimità mantenendo i punti dati non simili lontani. Nonostante la sua semplicità è uno tra i metodi più popolari e semplici utilizzati. In R è facilmente utilizzabile tramite la libreria `tfsknn`.

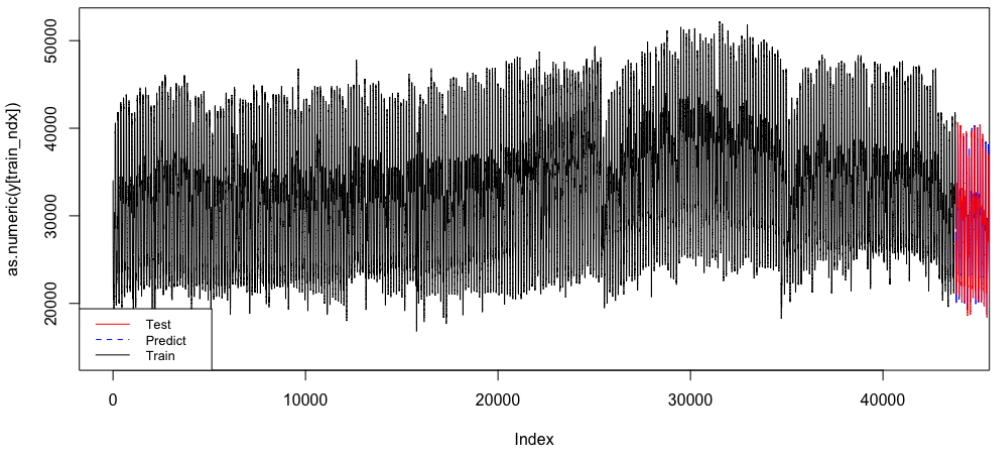
In quest'analisi il dataset è stato diviso in train e test, utilizzando come test l'ultimo mese disponibile, ed i parametri principali utilizzati per le previsioni sono:

- `K`: numero di Nearest Neighbors, in questo caso il numero di sequenze da considerare. Sono state effettuate diverse prove tra cui i valori 3, 5, 9, 12, 15. Si è riscontrato che i risultati migliori si sono ottenuti con il valore 12.
- `H`: numero di valori da prevedere (orizzonte previsivo). Per 30 giorni il valore impostato è 6*24*30.
- `Lags`: numero di lags da considerare (lookback) per effettuare le previsioni. Valore associato alla stagionalità, pertanto 1:6*24*7 per la stagionalità settimanale.
- `msas`: una stringa che indica la strategia Multiple-Step Ahead utilizzata quando viene previsto più di un valore. Può essere "`recursive`" o "`MIMO`" (Multi Input Multi Output). I risultati migliori si sono ottenuti con "`MIMO`".
- `cf`: indica la funzione di combinazione utilizzata per aggregare i target associati ai vicini più prossimi, può essere "median", "weighted" o "mean". Si sono ottenuti risultati migliori impostando "mean".
- `transform`: indica se i campioni di addestramento vengono trasformati e può assumere valori "additive" "multiplicative" oppure "none". Si sono riscontrati miglior risultati impostando "multiplicative".

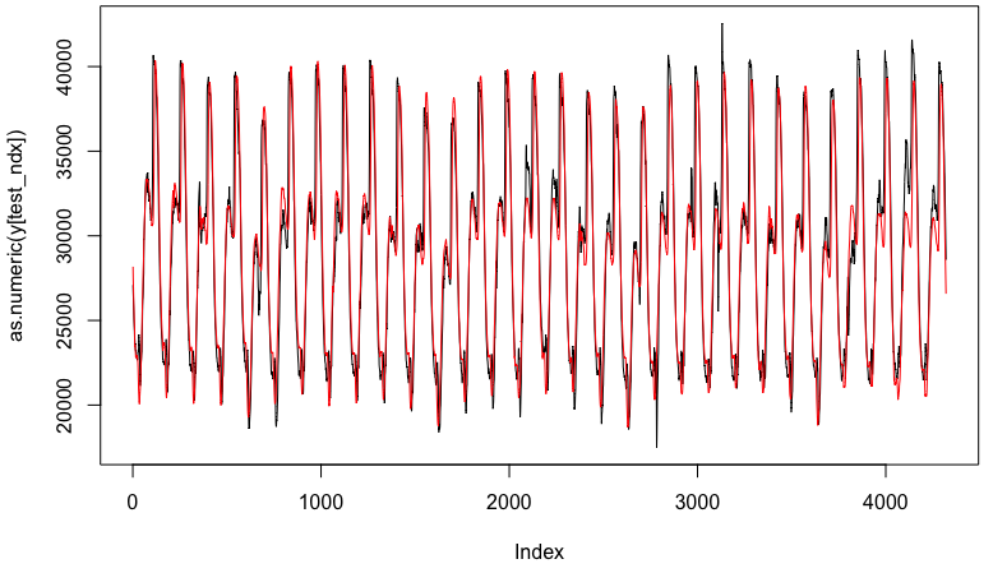
Nella tabella di seguito vengono mostrati 3 tentativi tra quelli effettuati:

Modello	**MAE**	**MAPE**
k = 7, msas = recursive, cf = median, transform = additive	1805.27	6.30
k = 7, msas = MIMO, cf = weighted, transform = multiplicative	1535.39	5.14
k = 12, msas = MIMO, cf = mean, transform = multiplicative	1354.37	4.57

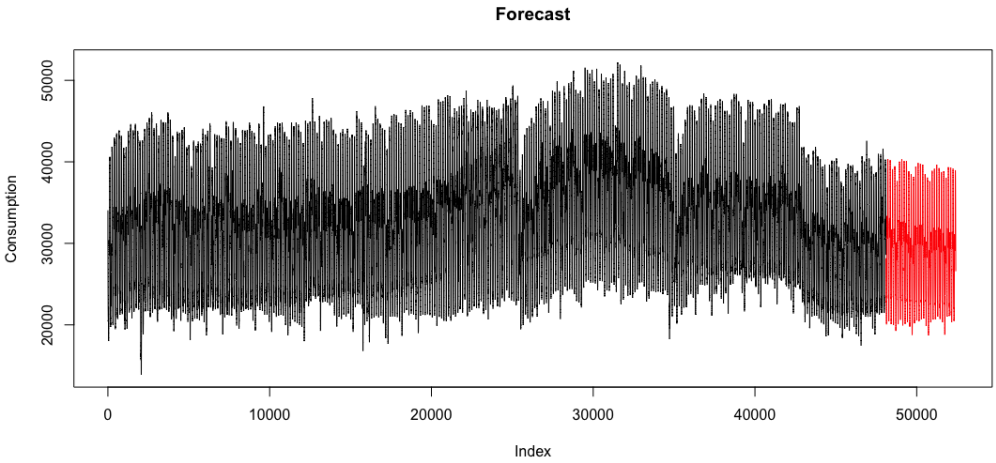
Come si può notare dalla tabella, è stato scelto il terzo modello. Di seguito vengono mostrate le previsioni per il mese di Novembre.



Di seguito viene mostrato il dettaglio delle previsioni per il mese di Novembre, in nero vengono mostrati i dati effettivi e in rosso le previsioni.



Nel grafico seguente, invece, vengono mostrate le previsioni richieste per il mese di Dicembre.



LSTM

LSTM (Long Short-Term Memory) è un'architettura basata su Recurrent Neural Network (RNN) ampiamente utilizzata nell'elaborazione del linguaggio naturale e nella previsione di serie temporali.

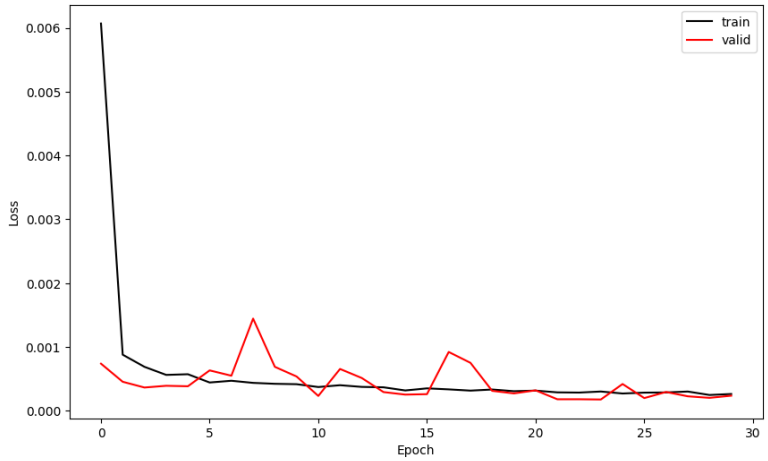
Come anticipato precedentemente, questa implementazione è stata effettuata utilizzando il linguaggio Python, in particolare la libreria `Keras` e l'ambiente di lavoro Google Colab.

Per poter utilizzare correttamente le funzioni rese disponibili da Keras, è necessario effettuare alcune operazioni di preprocessing sul dataset (oltre alla consueta suddivisione in train e test), ovvero:

- normalizzazione, viene infatti utilizzata la funzione `MinMaxScaler()` che normalizza i valori nel range 0, 1;
- rimodellazione, i dati vengono rimodellati con una finestra di "ritardi" pari a 6*24*7 (per rappresentare la stagionalità settimanale);
- reshape, il train set viene organizzato in una nuova forma, passando ad un array 3d con dimensioni (*sample*, *steps*, *features*).

Sono state effettuate diverse prove per il numero di neuroni, di epoche e la grandezza del batch size. Si è riscontrato che i parametri migliori per il modello è stato quindi una RNN con un layer LSTM da 60 neuroni, un layer di output di tipo Dense con un solo neurone in uscita, addestrata su 30 epoche e con batch size pari a 64. Più precisamente i valori simili a quelli scelti producevano un MAE e un MAPE molto simili tra loro nella fase di training e test ma le previsioni migliori sul mese di dicembre sono risultate più consone attraverso il modello selezionato (con altri paramentri i risultati erano fortemente distorti rispetto all'andamento generico della serie storica).

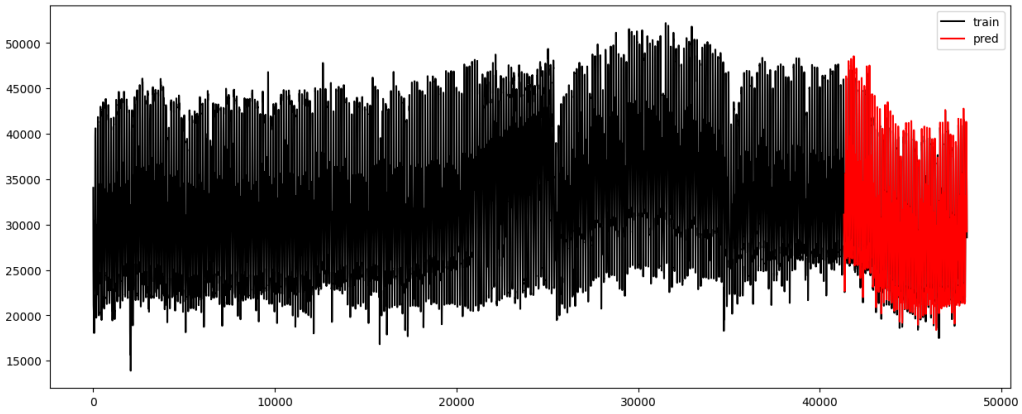
La loss function (impostata con il Mean Squared Error) prodotta nella fase di train della rete è visibile nel grafico successivo.



Si nota una stabilizzazione della loss function sia per il train che per il validation. Questo potrebbe esser sintomo di overfitting sui dati, ipotesi che viene aggravata dai valori di performances ottenuti sul test:

Modello	**MAE**	**MAPE**
LSTM con 60 neuroni, un layer Dense con un neurone, 30 epoche, batch size = 64	465.01	0.02

I risultati potrebbero essere confermati anche dalla visualizzazione delle previsioni poichè si nota un'elevata fedeltà rispetto ai sottostanti dati di train.

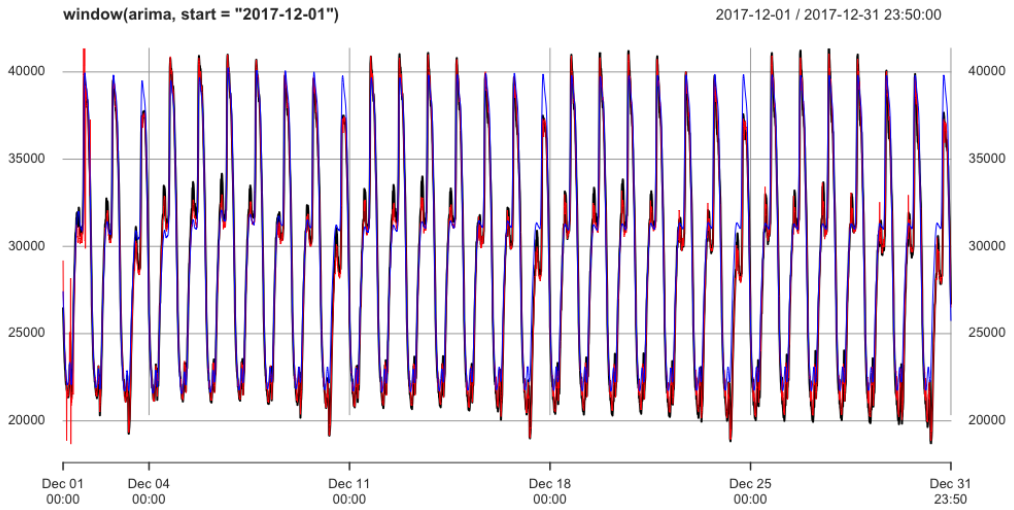


Conclusioni

A seguito delle analisi esposte, i modelli che hanno prodotto dei miglior risultati sono:

Modello	**MAE**	**MAPE**
ARIMA$(2, 1, 0)(0, 1, 1)_7$ con dummy	1031.17	3.65
Dummies + LLT + seasonal 7 dummy + seasonal 365 trigonometric 2 harmonics	1080.84	3.83
LSTM con 60 neuroni, un layer Dense con un neurone, 30 epoche, batch size = 64	465.01	0.02

Nel grafico sottostante vengono mostrate le predizione di questi tre modelli per il mese di Dicembre (in nero il modello ARIMA, in rosso il modello UCM e in blu il modello LSTM):



Dai risultati ottenuti sembrerebbe che il modello LSTM ottenga i risultati migliori ma rimane la perplessità che questo sia dovuto ad un overfitting sui dati di train. Probabilmente effettuando il train su più dati e modificando la configurazione della rete utilizzando un ulteriore numero di layer o di neuroni o inserendo delle regolarizzazioni si potrebbero ottenere dei risultati più affidabili per le previsioni.