Streaming Data Management and Time Series Analysis Project

Doci David
799647

Introduzione

Lo scopo del progetto è studiare una serie temporale per poter successivamente prevedere i 2 mesi successivi (Settembre 2020, Ottobre 2020), utilizzando tre modelli di forecast differenti:

- Modello ARIMA (Auto Regressive Integrated Moving Average).
- Modello UCM (Unobserved Components Model).
- Modello Recurrent Neural Networks (RNN).

Essi verranno confrontati utilizzando il Mean Absolute Error (MAE) come metrica di riferimento.

Dataset e Pre Processing

Il dataset fornito contiene valori orari per un periodo complessivo di 730 giorni (2 anni), il periodo di inizio è datato 1° Settembre 2018 mentre il termine della serie temporale risale al 31 Agosto 2020. Durante lo studio esplorativo dei dati forniti, si è constatato come non fossero presenti i valori relativi delle ore 3:00 dei giorni 31 Marzo 2019 e 29 Marzo 2020. Tale problema è dovuto al cambio dell'ora avvenuto proprio in quei giorni, come suggerito da coloro che hanno fornito il dataset i valori mancanti sono stati immessi prendendo i valori della serie storica precedenti. L'esiguita del periodo complessivo della serie storica in questione ha fatto sì che essa fosse suddivisa immettendo il 90% (657 giorni di osservazioni) dei dati disponibili nel training set e il

rimanente 10% (73 giorni di osservazioni) nel validation set. In seguito si è ritenuto necessario scalare i dati per poter diminuire il tempo necessario per ultimare i training dei vari modelli ed aumentare quindi l'efficienza di quest'ultimi.

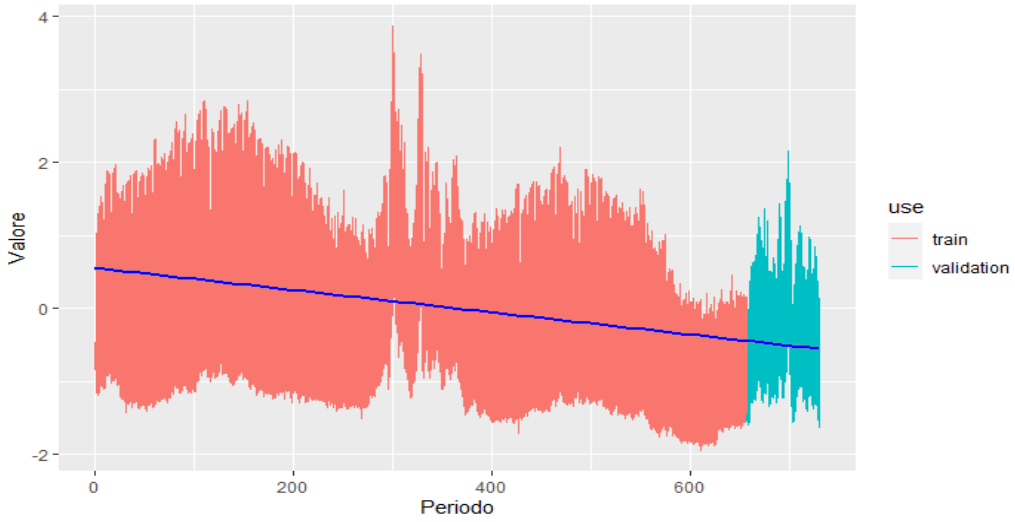

Figura 1: Serie Storica suddivisa in Train e Validation set

ARIMA

Per poter costruire ed in seguito individuare il miglior modello ARIMA si è seguita la procedura Box-Jenkins. La prima fase di questo metodo riguarda l'analisi preliminare nella quale si verifica la stagionalità della serie, mediante l'ausilio dell'analisi grafica dei plot ACF (Autocorrelation) e PACF (Partial autocorrelation function). Analizzando tali grafici si può notare come sia presente stagionalità giornaliera, la presenza è palesata da picchi costanti (ogni 24 lag) nel correlogramma ACF (Figura 2). Il passo successivo è stato applicare il secondo punto del metodo Box-Jenkins, ovvero l'identificazione del modello mediante l'individuazione degli ordine p,d,q del modello Arima non stagionale e i valori P,D,Q del modello stagionale ARIMA. Per far ciò si visionano i correlogrammi di Autocorrelazione (ACF) e di Autocorrelazione parziale (PACF). Osservando il PACF della serie storica si è deciso di costruire il primo modello ARIMA utilizzando un AR(3) non stagionale.

Come individuato precedentemente la serie storica studiata non è stazionaria in varianza, si è deciso quindi di differenziare i valori ogni 24 ore (Figura 3), fornendo al modello ARIMA una integrazione I(1).

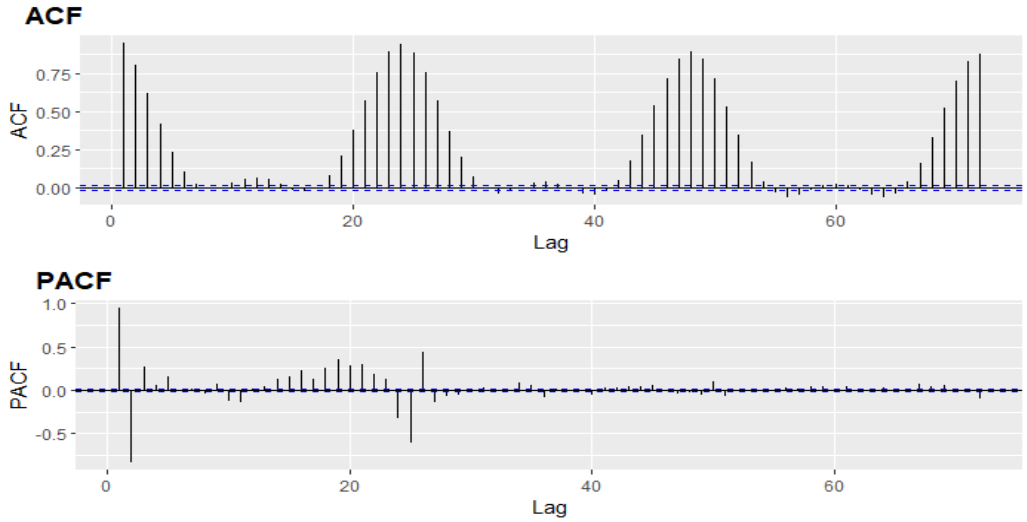

Figura 2: Grafico ACF/PACF della serie storica

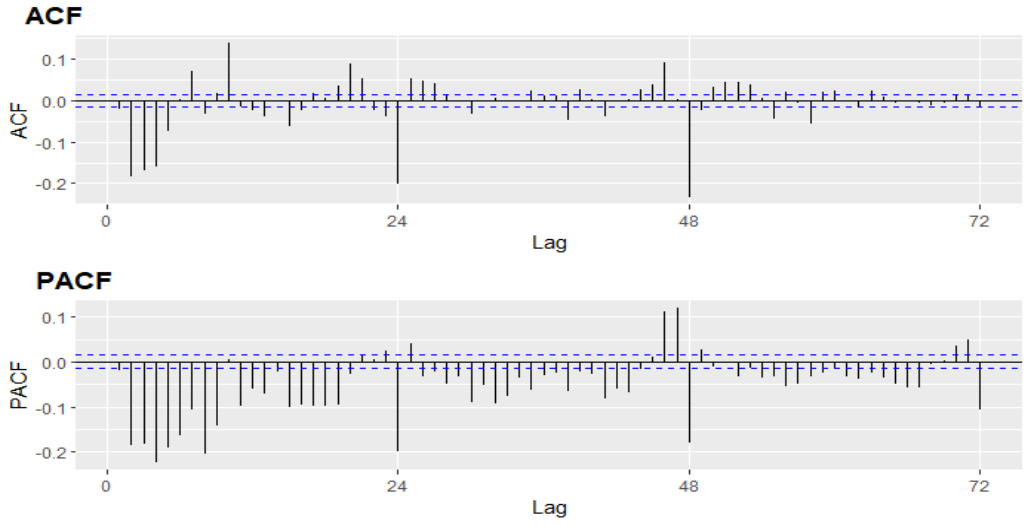

Figura 3: ACF e PACF della Serie Storica Differenziata

Dalle osservazioni precedenti si è quindi formato un modello SARIMA$(3,0,0)(0,1,0)_{24}$
.

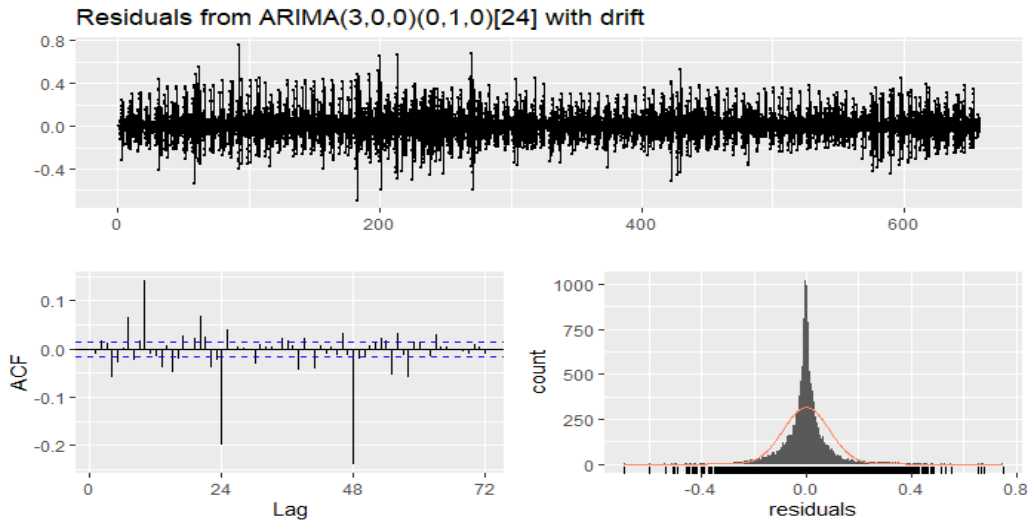

Figura 4: Residui del modello SARIMA$(3,0,0)(0,1,0)_{24}$

Osservando il correlogramma ACF prodotto dal modello, si possono riscontrare due picchi particolarmente elevati intorno al lag 24 e 48 (Figura 4), si è provato quindi a costruire un secondo modello SARIMA$(3,0,0)(0,1,2)_{24}$.

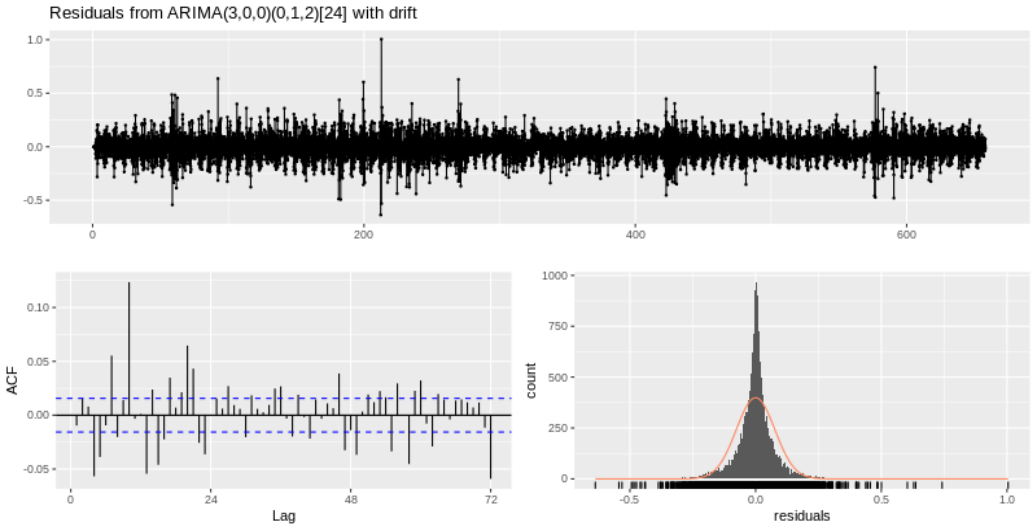

Figura 5: Residui del modello SARIMA$(3,0,0)(0,1,2)_{24}$

Successivamente si sono provati differenti modelli SARIMA che potessero migliorare quelli già presentati, ma nessuno di quelli testati è risultato migliore. Si è provato ad utilizzare la tecnica del Grid Search mediante l'utilizzo della funzione auto.arima presente nel pacchetto *forecast*. Tale funzione permette di trovare il modello ARIMA univariato migliore mediante la combinazione di tutti i parametri immessi all'inizio della ricerca (in questo caso dopo svariate prove si è deciso di lasciare i parametri standard, ove gli ordini massimi sia per la parte stagionale che non, fossero massimo pari a 5). Tale ricerca ha portato allo sviluppo del modello SARIMA$(5,0,0)(2,1,0)_{24}$.

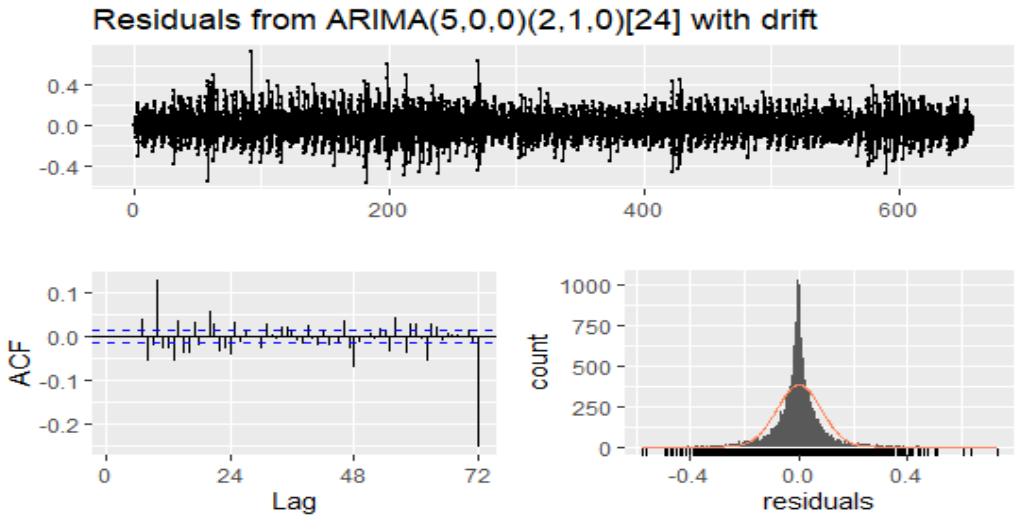

Figura 6: Residui del modello SARIMA$(5,0,0)(2,1,0)_{24}$

Come si può osservare dal correlogramma ACF (Figura 6), vie è un miglioramento rispetto ai precedenti correlogrammi poichè buona parte dei valori rientra dentro le bande, il risultato in termini di MAE però ha evidenziato come il modello prodotto mediante grid search fosse molto simile al primo modello SARIMA provato.

Modelli SARIMA	$(3,0,0)(0,1,0)_{24}$	$(3,0,0)(0,1,2)_{24}$	$(5,0,0)(2,1,0)_{24}$
Training	0.0536	0.0503	0.0529
Validation	0.4629	0.5128	0.4644

Tabella 1: Confronto dei risultati ottenuti dai modelli ARIMA

Dopo aver osservato i risultati prodotti si è deciso di selezionare il modello SARIMA $(3,0,0)(0,1,0)_{24}$ come miglior modello per questa metodologia.

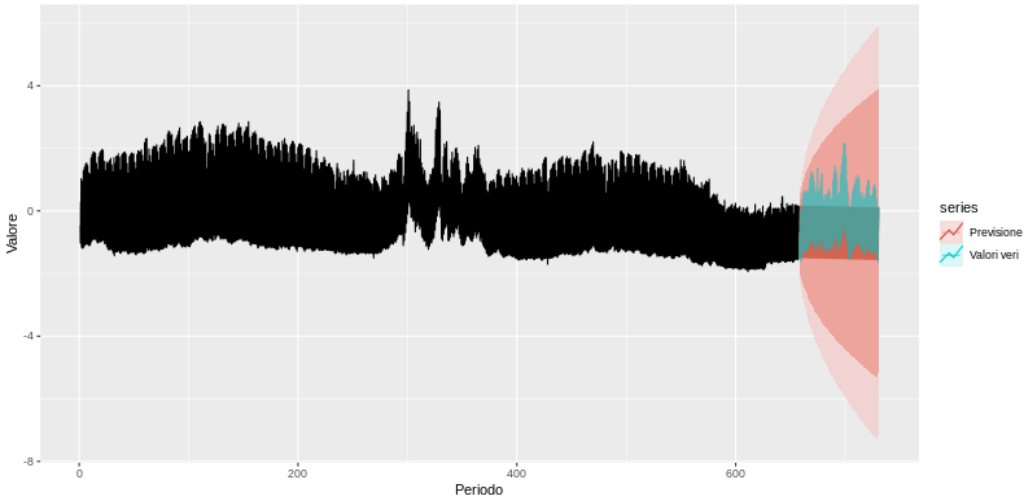

Figura 7: Predizione del modello SARIMA$(3,0,0)(0,1,0)_{24}$

UCM

Usando le considerazioni apprese mediante la costruzione dei modelli ARIMA e le informazioni acquisite nel pre processing si è deciso di costruire i seguenti modelli:

- Local Linear Trend con una componente stagionale giornaliera stocastica dummy.
- Local Linear Trand con stagionalità giornaliera trigonometrica
- Local Linear Trend con stagionalità Dummy e ciclo Annuale.

La validazione dei modelli UCM è stata eseguita mediante la predict di SSModel. Tale funzione sfrutta il framework KFAS che gli permette di gestire in maniera automatica le modifiche necessarie alle strutture SSModel per effettuare delle previsioni mediante l'utilizzo della forma state space senza dover ricalcolare i dati all'interno di quest'ultima.
I risultati prodotti dai modelli UCM sono:

Modelli UCM	$LLT + S.Dummy$	$LLT + S.trig$	$LLT + S.Dummy + C.Annuale$
Training	0.0100	0.0532	0.0010
Validation	0.4629	0.5335	0.3988

Tabella 2: Confronto dei risultati ottenuti dai modelli UCM

Si denota come il miglior modello UCM sia stato il local linear trend con stagionalità dummy e ciclo annuale, riportando una validation loss pari a 0.3988. La predizione prodotta è risultata:

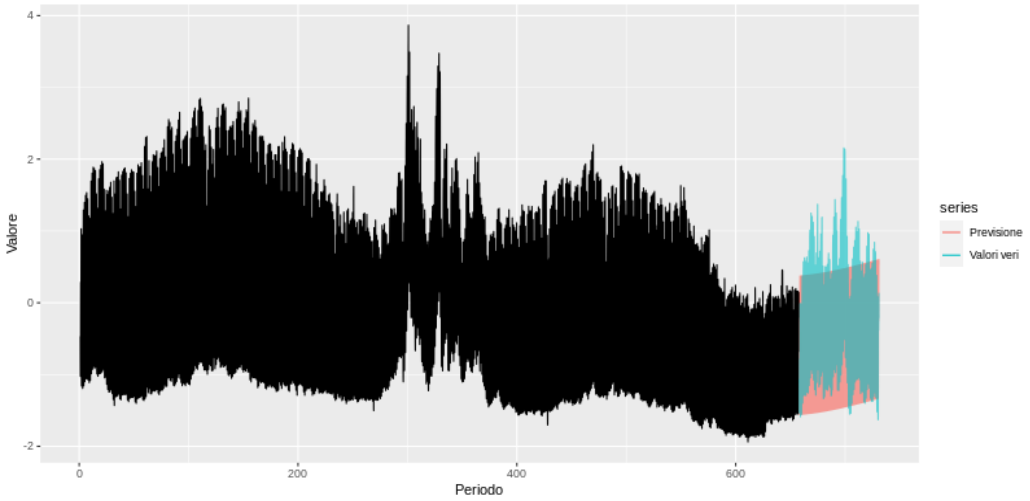

Figura 8: Predizione del miglior modello UCM

Modelli di Machine Learning

I modelli di machine learning utilizzati sono stati:

- Modello LSTM (Long Short-Term Memory).
- Modello RNN (Recurrent Neural Network).
- Modello GRU (Gated Recurrent Unit).

Anche in questo caso il training e il test (o validation) set utilizzati per l'apprendimento dei modelli di machine learning hanno una struttura simile a quella dei precedenti set, per far si che si potessero confrontare con maggior facilità tutti i modelli addestrati. In maniera similare i dati sono stati normalizzati per avere media 0 e varianza 1 utilizzando la funzione MinMaxScaler di *sklearn*, in questo caso però la standardizzazione è essenziale per la costruzione degli algoritmi, dato che i modelli di machine learning richiedono un input avente determinate caratteristiche. Per quanto concerne la lunghezza della sequenza si è deciso, dopo svariate prove, di porre tale fattore pari a 336.

Tutti e tre i modelli sviluppati hanno avuto come ottimizzatore "Adam", mentre come funzione di perdità è stata utilizzata ovviamente il mean absolute error. Le strutture dei tre modelli sono tutte riconducibili a:

- Layer di input dei relativi modelli con un numero di neuroni pari a 64 con activation = "tanh".
- Dropout layer con valore 0.2
- Layer di input dei relativi modelli con un numero di neuroni pari a 64 con activation = "tanh".
- Dropout layer con valore 0.2
- Layer di input dei relativi modelli con un numero di neuroni pari a 64 con activation = "tanh".
- Dropout layer con valore 0.2
- Dense layer con valore 1

L'addestramento è stato eseguito in 5 epoche, scelto dopo svariate prove per trovare il miglior tradeoff per tutti e tre i modelli, mentre la batch size utilizzata è pari a 128. Le valutazioni dei modelli di machine learning sono state rappresentate nella seguente tabella:

Modelli ML	LSTM	RNN	GRU
Training	0.0492	0.0349	0.0319
Validation	0.0427	0.0184	0.0176

Tabella 3: Confronto dei risultati ottenuti dai modelli ML

Il miglior modello è risultato il Gated Recurrent Unit (GRU) con un Validation loss pari a 0.0176, di seguito si mostrano le predizioni del modello in questione:

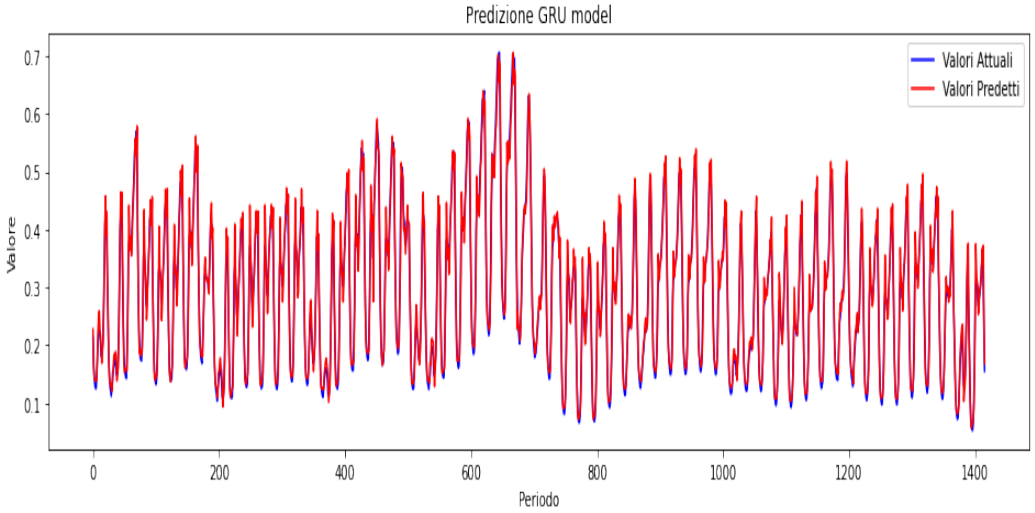

Figura 9: Predizione modello GRU

Conclusioni

I migliori risultati delle tre metodologie sono:

Model	ARIMA	UCM	ML
Training	0.0536	0.0010	0.0319
Validation	0.4629	0.3988	0.0176

Tabella 4: Confronto dei risultati ottenuti dai migliori modelli

Il miglior modello è risultato il Gated Recurrent Unit (GRU) relativo alla metodologia del machine learning, esso si è contradistinto grazie ad una MAE, relativa al validation set, estremamente bassa rispetto ai modelli lineari, riuscendo a cogliere sia la maggior parte del trend che della stagionalità indipendentemente dagli input dati dall'utente. Al contrario le precedenti metodologie sono risultate più sensibili all'input immessi rendendole così maggiormente ricettive all'errore umano.