Università degli Studi di Milano-Bicocca **Alessandro Riboni**
M.Sc. Data Science a.riboni2@campus.unimib.it
June 11, 2020 Matricola: 847160

Streaming Data Management and Time Series Analysis

Introduzione

In questo report vengono presentate tre metodologie per risolvere un problema di forecasting di prezzi del mercato energetico. In particolare, vengono applicate le tecniche studiate durante il corso di *Streaming Data Management and Time Series Analysis*, effettuando un confronto tra tre diversi modelli: un modello ARIMA, un modello UCM e un modello predittivo non-lineare (GRU). Tali modelli sono stati fittati su un *training_set* contenente i prezzi giornalieri dal 1 Gennaio 2010 al 31 Dicembre 2018. L'ultimo anno è stato inizialmente utilizzato come *validation_set* e, solo dopo aver scelto le configurazioni migliori, è stato utilizzato per ottenere le previsioni finali dal 1 Gennaio 2019 al 30 Novembre 2019.

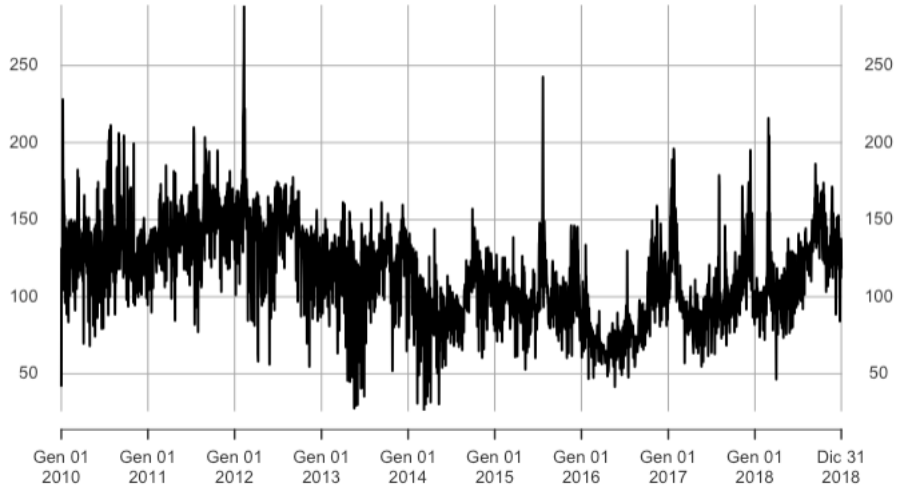

Figure 1: Prezzi energetici dal 1 Gennaio 2010 al 31 Dicembre 2018

La misura di errore scelta per confrontare i modelli è il **MAPE** (*Mean Absolute Percentage Error*). Questa misura risulta essere facile da interpretare e permette di avere un indicazione su quanto si sta sbagliando. L'utilizzo di tale misura è possibile perchè i valori dei prezzi sono sempre maggiori di zero. Inoltre, il MAPE viene sempre riportato con un grafico in cui vengono confrontati i valori reali con i valori predetti in modo da avere anche un riscontro visivo della bontà delle previsioni.
Nelle sezioni successive vengono riportate le analisi effettuate per le tre diverse metodologie e, infine, viene riportato un confronto tra di esse e le considerazioni finali.

Trasformazione della serie temporale

Come suggerisce la procedura di Box-Jenkins, il primo punto fondamentale è quello di sistemare la stazionarietà in varianza. Non essendo presente una forte relazione lineare tra le media e la deviazione standard della serie storica, è stata preferita una trasformazione automatica della serie tramite la funzione *BoxCox* con il parametro *lambda="auto"* piuttosto che effettuare una trasformazione logaritmica. I modelli sono stati fittati sulla serie temporale trasformata e la trasformazione inversa è stata effettuata prima di salvare le previsioni dei tre modelli sul *test_set*.

Modelli ARIMA

Dopo aver sistemato la stazionarietà in varianza sono stati analizzati i grafici ACF e PACF della serie con due *lag.max* diversi in modo da cogliere e analizzare sia gli andamenti a breve che a lungo termine.

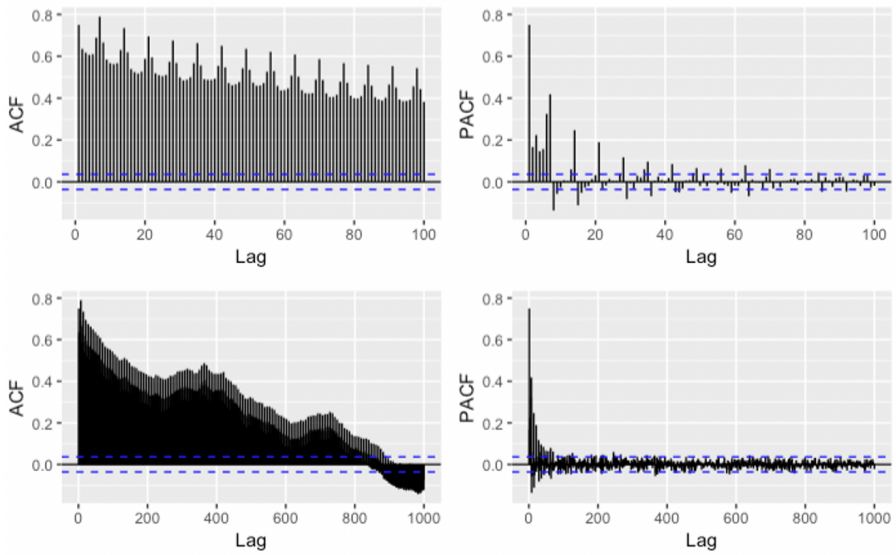

Figure 2: ACF e PACF sui dati del *training_set*.

Dai grafici ACF si può notare una discesa lineare e la presenza di stagionalità settimanale e annua, mentre da PACF si può notare una discesa geometrica ogni sette giorni. Inoltre, dai grafici sembra essere presente un trend, nei primi anni decrescente e in quelli successivi crescente.
Per tali motivi sono state inizialmente effettuare le due integrazioni e i grafici dei residui hanno suggerito la necessità di inserire le componenti AR(1)MA(1) e una componente SMA(7). Per ridurre il numero di lag superiori alle bande è stata inserito una componente SAR(7). Nonostante ciò, erano presenti ancora dei lag superiori alle bande, i quali suggerivano di aumentare l'ordine di AR e considerare anche la stagionalità annua. Per tali motivi è stato effettuato un Grid Search con l'obiettivo di identificare il modello con il valore dell'AIC minore facendo variare il coefficente p di AR nel range [1,6] e il numero di termini di Foureir k da inserire come regressori per cogliere la stagionalità annua. I valori considerati per k sono stati [1,5,10].
Il modello migliore è risultato essere ARIMA(6,1,1)(1,1,1)[7] con $k = 10$, il quale ha ottenuto un MAPE di 9.00% sul *training_set* e di 10.94% sul *validation_set*.

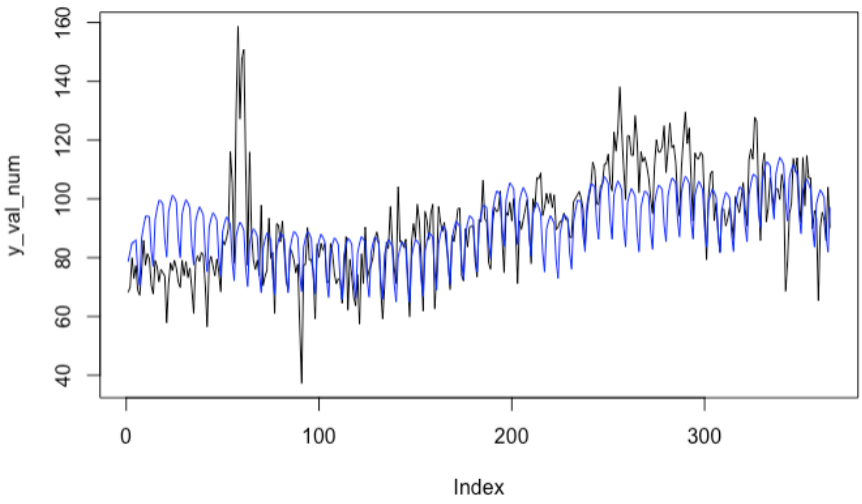

Figure 3: Previsione del modello ARIMA sui dati di validation.

Infine, si è deciso di provare ad aggiungere delle variabili dummy per evidenziare le festività che non cadono di Domenica in modo da cogliere maggiore informazione. Inizialmente tale aggiunta al modello non ha portato grandi benefici in termini di performance ma, dopo aver fatto variare il numero di termini di Fourier si è ottenuto un modello migliore. In particolare, con l'aggiunta delle dummy per le festività e con $k = 1$ si è ottenuto un MAPE del 8.53% sul *training_set* e di 10.08% sul *validation_set*.
Nella figura seguente viene riportato il confronto tra il modello ARIMA senza dummy con k = 10 e il modello con dummy e k = 1 sui dati presenti nel *validation_set*.

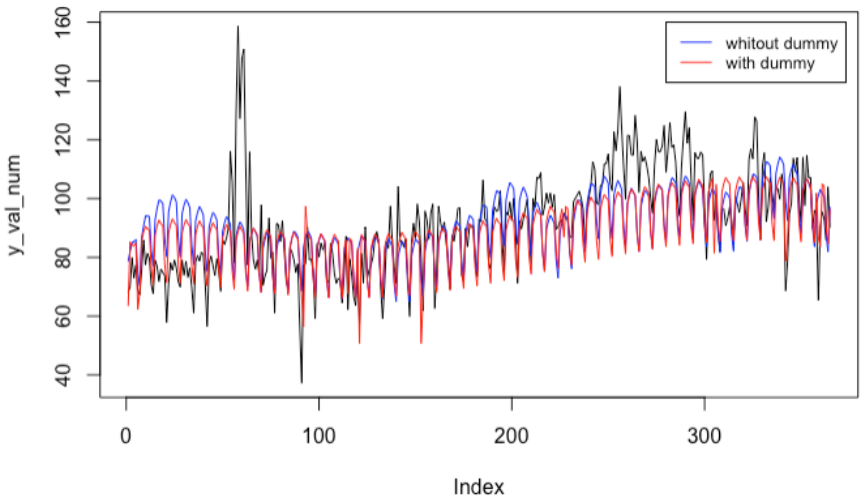

Figure 4: Confronto tra modelli ARIMA sui dati di validation.

I residui di entrambi i modelli sembrano essere distributi come una normale e nei grafici ACF e PACF risultano esserci pochi picchi superiori alle bande. Nonstante ciò, il test di Ljung-Box rifiuta l'ipotesi nulla secondo cui i residui sono distributi in modo indipendente, sembra perciò esserci ancora una correlazione seriale. Considerando le performance migliori sul *validation_set*, è stato scelto il modello ARIMA(6,1,1)(1,1,1)[7] con k = 1 e le dummy per le festevità.

Modelli UCM

Durante lo sviluppo del modello a componenti non osservabili sono state tenute in considerazione le analisi preliminari effettuate sulla serie storica. Di conseguenza si è deciso di sviluppare due modelli. Il primo composto da un Random Walk per catturare il trend, una stagionalità settimanale con dummy stocastica e una stagionalità annuale trigonometrica stocastica. Il secondo, invece, è composto da un Random Walk Integrato, una stagionalità settimanale con dummy stocastica e una stagionalità annuale trigonometrica stocastica. Per entrambi i modelli si ha effettuato un Grid Search sul numero delle armoniche per cogliere la stagionalità annua in modo da ottimizzare il MAPE sul validation set.
Si è scelto di usare la libreria *KFAS* e in particolare la funzione *SSModel*, la quale permette di costruire il modello come combinazione delle diverse componenti. Per evitare che venissero stimate varianze per ogni diverso parametro è stata riscritta la funzione update in modo che le varianze delle sinusoidi fossero tutte uguali. I valori del numero di armoniche considerati sono [2,6,10,12].
Il modello che ha ottenuto le performance migliori è stato quello composto dal RW, le due stagionalità e $k = 14$, raggiungendo un MAPE di 7.30% sul *training_set* e 15.19% sul *validation_set*.

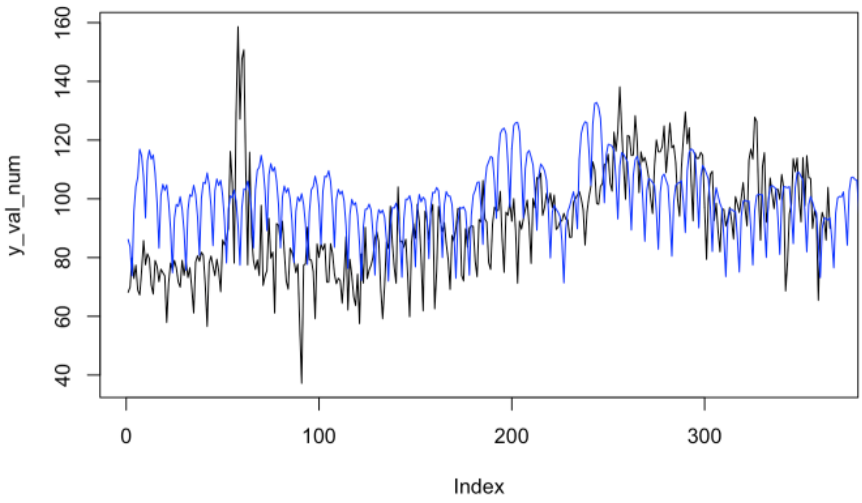

Figure 5: Previsione del modello UCM sui dati di validation.

Infine, si ha provato ad inserire le dummy per le festività utilizzate come regressori nei modelli ARIMA ottenendo però dei risultati peggiori. Per tale motivo si è deciso di non inserirle nel modello UCM finale.

Modelli non lineari (GRU)

L'ultima tipologia di modelli considerata è quella dei modelli non lineari, in particolare è stata implementata una rete neurale ricorrente tramite lo sviluppo di un architettura GRU (Gated Recurrent Unit). Il vantaggio proposto da questo tipo di rete è quello di poter modellare un comportartamento dinamico tramite le informazioni degli istanti precedenti e, di conseguenza, risulta particolarmente adatta a gestire dati di serie temporali.
La serie storica è stata trasformata in modo da ricondursi ad un problema di *supervised learning*, ovvero ogni elemento della serie è stato associato ai suoi N ritardi, i quali vengono utilizzati

come regressori per prevedere tale valore. Dopo diversi test è stato scelto di utilizzare $N = 365$ in modo da avere sempre una visione annuale e, allo stesso tempo, un numero adeguato di dati su cui allenare la rete.
In particolare è stata implementata una **stacked GRU** composta da due layer GRU con rispettivamente 64 e 128 neuroni. In entrambi i layer ricorrenti sono stati utilizzati *dropout* e *recurrent_dropout* con rispettivamente i valori di 0.1 e 0.5 in modo da limitare l'overfitting. Infine, è presente il layer Dense di output formato da un solo neurone con funzione di attivazione sigmoidale. La funzione di perdità utilizzata per la fase di training è stata il *mean square error* ed è stato scelto *Adam* come ottimizzatore.
Per poter confrontare i risultati ottenuti con i modelli precedenti le previsioni sul *validation_set* vengono effettuate un passo in avanti ma, ad ogni iterazione, l'elemento predetto viene aggiunto alla coda dei regressori per predire l'elemento successivo.
Dopo 20 epoche, è stato raggiunto un MAPE di 9.60% sul *training_set* e 10.87% sul *validation_set*. Come si può notatre dalla figura seguente la rete ricorrente è in grado sia di gestire la stagionalità settimanale che di cogliere l'andamento annuale dei prezzi presenti nel *validation_set*.

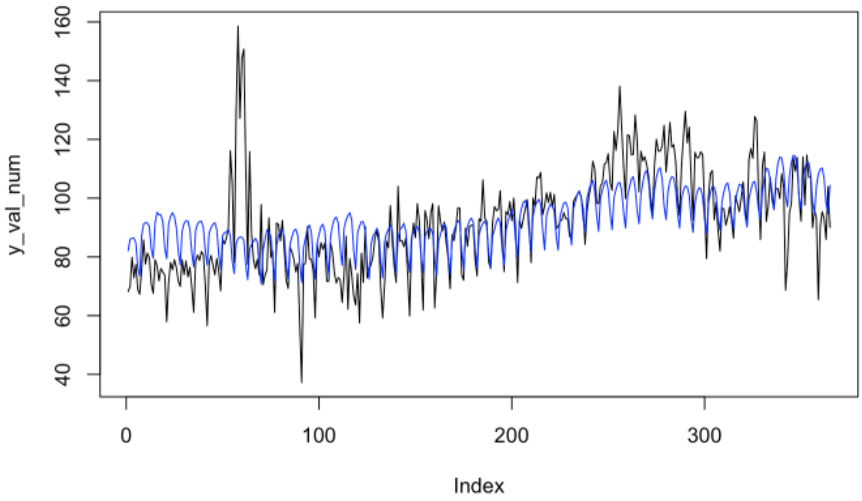

Figure 6: Previsione della rete neurale GRU sui dati di validation.

Considerazioni finali

Nella tabella seguente vengono riportate le performance misure tramite il MAPE sui tre diversi modelli implementati. Come si può notare il modello migliore risulta essere il modello ARIMA. L'inserimento delle dummy per le festività ha portato ad un aumento nelle performance riuscendo a cogliere determinati picchi nel consumo energetico dovuti a giorni di vacanza.

Modello	**MAPE training_set**	**MAPE validation_set**
ARIMA	8.53%	10.08%
UCM	7.30%	15.19%
GRU	9.60%	10.87%

In conclusione, i modelli ARIMA e UCM sono stati riallenati nuovamente su tutti i dati disponbili (*training_set* e *validation_set*) mentre per la rete GRU si ha scelto di partire dai pesi ottenuti dalla fase di training e allenare nuovamente il modello solo sui dati presenti nel *validation_set*

per 5 epoche. Successivamente vengono riportate le previsioni dei tre modelli sul *test_set*, ovvero su un arco temporale di 334 giorni dal 1 Gennaio 2019 al 30 Novembre 2019.

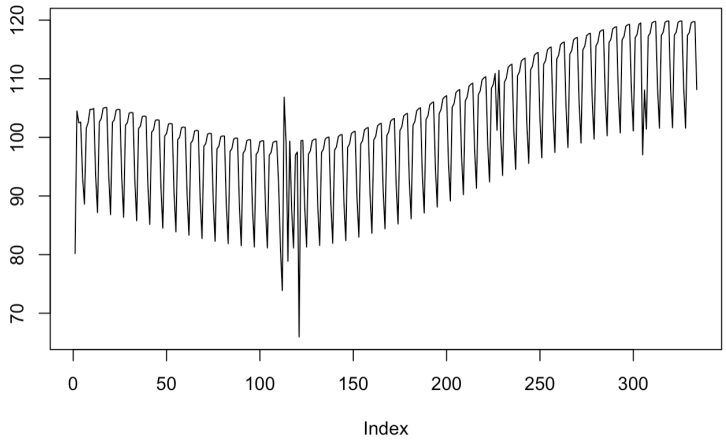

Figure 7: Previsione del modello ARIMA sui dati di test.

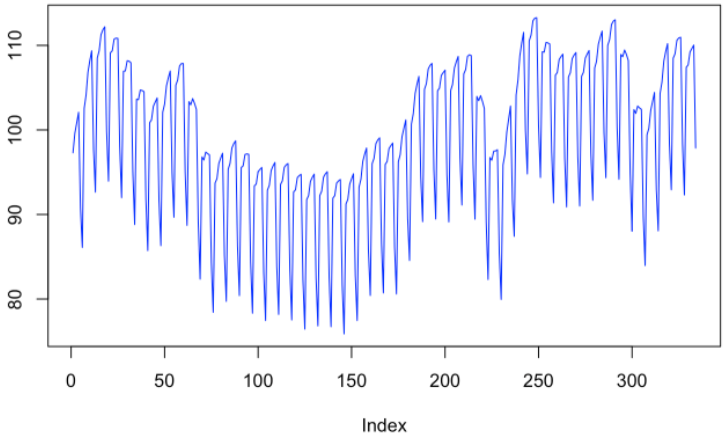

Figure 8: Previsione del modello UCM sui dati di test.

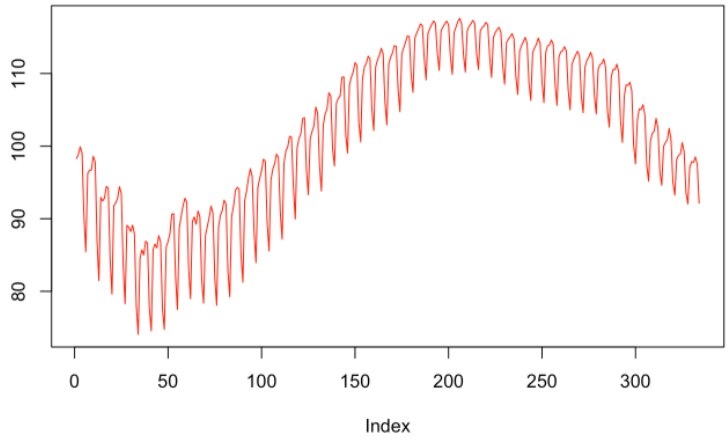

Figure 9: Previsione del modello GRU sui dati di test.