

UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

STREAMING DATA MANAGEMENT AND TIME SERIES
ANALYSIS

PROGETTO FINALE

Forecasting time series:

Definizione, sviluppo e validazione di un sistema per
prevedere i prezzi giornalieri di un mercato energetico

Autore:

Matteo Gaverini - 808101 - m.gaverini1@campus.unimib.it

Settembre 2020



Sommario

Realizzare un sistema utile a prevedere una qualche quantità o prezzo, risulta fondamentale per le imprese in relazione ai vari obiettivi di business che intendono perseguire. La maggior parte delle aziende dei più disparati settori (finanza, meccanica, energia etc.), impiegano costantemente modelli predittivi al fine di massimizzare i ricavi e/o minimizzare i costi di produzione. Il progetto si focalizza sulla previsione di 11 mesi dei prezzi giornalieri di un mercato energetico avendo a disposizione una serie temporale giornaliera in cui sono registrati in 9 anni, i prezzi storici di tale mercato. Le previsioni vengono effettuate realizzando rispettivamente tre modelli: i primi due provenienti dal mondo statistico che risultano essere *ARIMA* e *UCM*, l'altro invece proveniente dal mondo *Machine Learning* ovvero *RNN*. Questi sistemi, una volta addestrati, vengono poi validati su un unico *test set* per analizzare e confrontare le diverse performance in modo da decretare alla fine il modello migliore.

1 Introduzione

Una serie temporale o in inglese *time series* è definita come un insieme di osservazioni misurate sequenzialmente nel tempo [1]. Più precisamente è una sequenza finita di valori di variabili casuali equispaziati in cui esiste tra di loro una relazione di dipendenza. Le serie temporali, se sfruttate efficacemente, risultano essere di grande valore in ambito aziendale: grazie a questa tipologia di dati, le imprese non solo possono svolgere analisi approfondite riguardo per esempio lo storico dei prodotti venduti ma soprattutto possono realizzare sistemi in grado di prevedere le vendite future di un prodotto o le domande di un certo servizio. Esistono molteplici casi di aziende che, servendosi di modelli predittivi affidabili, massimizzano i ricavi e/o minimizzano i costi. Basti pensare per esempio alla grande distribuzione organizzata che, attraverso dei sistemi di previsione basati sul *Machine Learning*, riesce ad ottimizzare la catena logistica e a controllare accuratamente le scorte [2]. Oltre a questo settore, si trovano anche altre aree di mercato dove tali modelli vengono utilizzati come supporto alle decisioni di business. Un caso concreto è il settore energetico che ha adottato con molta convinzione l'analisi predittiva per diverse finalità tra i quali la previsione dei fabbisogni futuri e il miglioramento delle prestazioni complessive [3]. Il progetto si incentra sulla previsione di 11 mesi (dal 1/1/2019 al 30/11/2019) dei prezzi giornalieri di un mercato energetico avendo a disposizione una *time series* giornaliera in cui sono registrati in 9 anni (dal 2010 al 2018) i prezzi storici di tale mercato. Per perseguire tale obiettivo, si sono realizzati tre modelli, nello specifico *ARIMA*, *UCM*, *RNN* e tutte le previsioni ricavate da ciascuno, sono ottenute adottando la strategia diretta o "one shot", cioè utilizzando

esclusivamente i dati di training si producono direttamente gli h valori predittivi. L'implementazione di ogni modello è avvenuto seguendo lo stesso procedimento, ovvero a partire da un *vanilla model* (modello base) si è cercato, attraverso continue operazioni di *refine*, di ottenere un sistema sempre più performante rispetto a quelli della stessa "famiglia" ricavati precedentemente. Per tutti i sistemi predittivi è stata effettuata l'ottimizzazione dei principali iperparametri utilizzando metodi differenti, nello specifico per *ARIMA* e *UCM* si è adottata una *grid search* mentre per *RNN* si è utilizzato *AutoML* (processo che ottimizza una funzione obiettivo determinando il valore ottimo degli iperparametri). Una volta addestrati i modelli con le configurazioni ottime degli iperparametri, si è poi proceduto alla loro validazione su un unico *test set* utilizzando come misura di errore il *MAPE* per decretare alla fine il sistema predittivo migliore.

2 Dataset

La serie temporale è costituita da 3287 prezzi storici giornalieri di un mercato energetico, registrati dal 1/1/2010 al 31/12/2018 (si veda Figura 1). La provenienza di tali dati è sconosciuta quindi non si sa quale sia effettivamente il mercato di riferimento (italiano, americano etc.). Osservando la serie temporale (si veda Figura 1), si evidenzia un trend stocastico visto che l'andamento non si ripete allo stesso modo nei vari istanti di tempo t (crescente nei primi due anni, decrescente dal 2012 al 2016 infine crescente fino al 2018). La presenza di tale trend è visibile anche dalla decomposizione della serie temporale nei tre elementi *trend*, *seasonal* e *remainder* (si veda Figura 2); ciò implica che la *time series* non risulta essere stazionaria in media. Oltre a questo, dalla decomposizione emerge anche un altro aspetto interessante ovvero la presenza della stagionalità annuale. Questo può essere legato al fatto che nei mesi invernali il prezzo della corrente elettrica e metano sia più alto rispetto ai mesi estivi visto il fabbisogno energetico diverso di case e uffici. Oltre alla stagionalità annuale, nei dati si riscontra anche una stagionalità settimanale, motivata dalla presenza di una discesa geometrica ogni 7 ritardi all'interno dei grafici *ACF* e *PACF* ottenuti dalla *time series* (si veda Figura 3). La presenza di tale stagionalità può essere condizionata dal fatto che nei giorni feriali il prezzo dell'energia elettrica sia più alto rispetto ai giorni festivi visto che sono aperte fabbriche, uffici e quindi il consumo energetico è nettamente superiore rispetto a quello consumato dalle città nei weekend. In ultima battuta osservando la serie temporale, si evidenziano anche dei picchi candidati ad essere *outlier* come ad esempio quelli registrati ad inizio del 2012 e metà del 2015. Questo può essere legato al fatto che forse in quei giorni sono avvenuti degli eventi particolari che hanno influenzato in maniera drastica i prezzi del mercato energetico però, non conoscendo

l'effettiva provenienza dei dati, non è stato possibile risalire a quale fosse la reale causa di tale situazione.

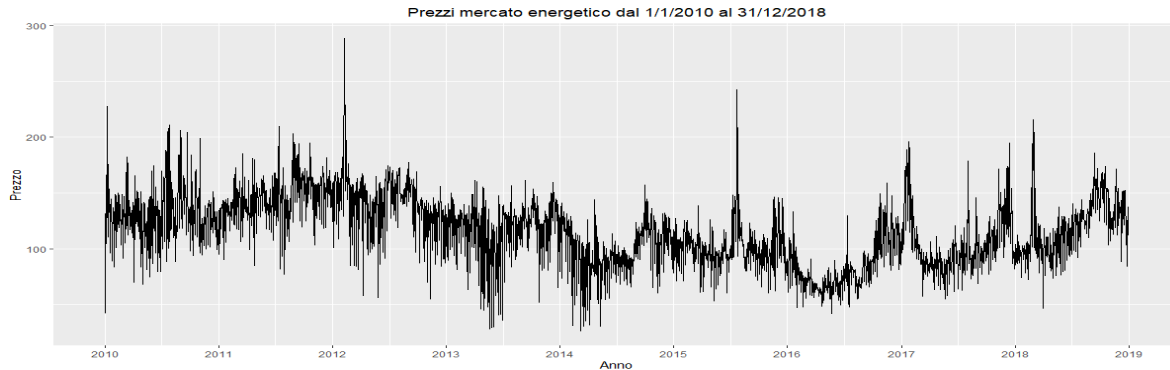


Figura 1: Time series prezzi

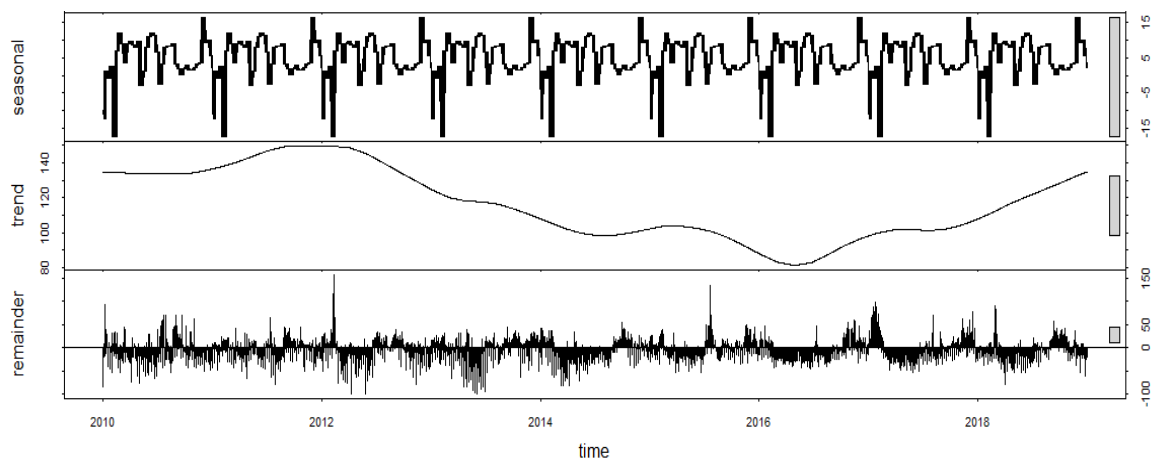


Figura 2: Decomposizione time series

3 Training, Test e Validation

Il dataset viene diviso in 3 parti: *training set* 78%, *validation set* 11% e *test set* 10%. Il *training set* impiegato per addestrare i modelli, include i primi 7 anni completi (dal 2010 al 2016), il *validation set* utilizzato per ottimizzare gli iperparametri, contiene tutti i dati del 2017 infine il *test set* usato per valutare i sistemi predittivi, include i

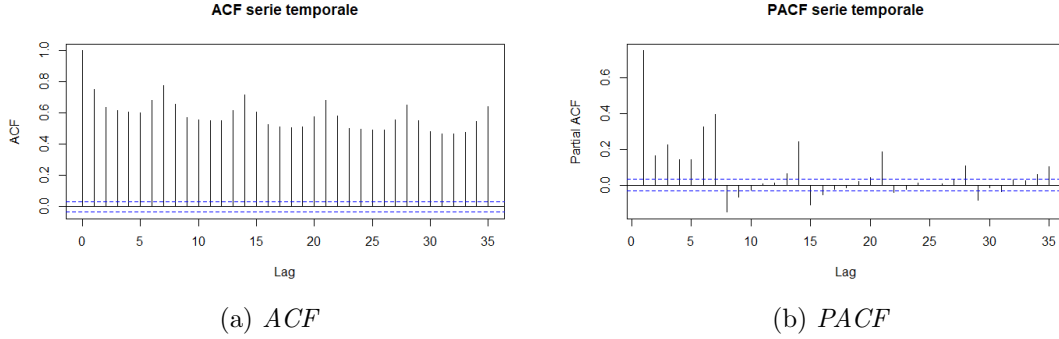


Figura 3: ACF e PACF time series

primi 11 mesi (334 giorni) del 2018. Si è deciso di non considerare tutto l'anno per il *test set* perché l'obiettivo del progetto è prevedere 11 mesi e non un anno completo di conseguenza si è ritenuto che valutare i modelli solamente su quell'orizzonte temporale fosse la soluzione più adeguata.

4 Misura di errore

Come si è detto nel Capitolo 1, la misura di errore utilizzata per valutare i modelli è il *MAPE* (acronimo di *Mean Absolute Percentage Error*). Matematicamente tale metrica è così definita:

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right| \times 100 \quad (1)$$

dove A_t è il valore reale e F_t è il valore predetto.

Si è scelto il *MAPE* e non la principale alternativa che è *RMSE* fondamentalmente per due motivi. Il primo è che tale metrica è più facile da interpretare rispetto all'altra mentre il secondo è che il *MAPE* si rivela più affidabile del *RMSE* in quanto, secondo una [survey](#) del 2013, la seconda metrica a differenza della prima, può essere condizionata dai dati su cui viene effettuato il calcolo e quindi cambiando il *test set*, si potrebbero ottenere valori completamente differenti rispetto a quelli ottenuti precedentemente.

5 Modelli

5.1 ARIMA

Il primo sistema realizzato è *ARIMA* (acronimo di *AutoRegressive Integrated Moving Average*), una classe di modelli statistici utilizzata per analizzare e svolgere previsioni. Il modello, in generale, è caratterizzato da 3 componenti così definite: *AR*, *I* e *MA*. Il primo elemento rappresenta un processo autoregressivo, il secondo indica l'eventuale differenziazione effettuata alla serie temporale per renderla stazionaria infine il terzo elemento rappresenta un processo a media mobile [4]. Queste componenti possono essere sia stagionali che non stagionali e in base a questo, il modello può essere *non-seasonal ARIMA* oppure *seasonal ARIMA* (*SARIMA*).

5.1.1 Procedimento

Per realizzare il sistema predittivo si è utilizzato il metodo *Box-Jenkins*, una procedura iterativa che consente di individuare il modello *ARIMA* più opportuno in base ai dati a disposizione. La prima operazione che si è effettuata è stato rendere stazionaria la serie temporale, condizione necessaria affinché il modello sia valido. Per far ciò si è applicata una trasformazione *Box-Cox* logaritmica, dopodiché per eliminare il trend e la stagionalità, si è effettuata sia una differenziazione stagionale settimanale che una differenziazione semplice. A questo punto, per verificare se la stazionarietà è stata raggiunta, si è svolto il test di *Augmented Dickey-Fuller* da cui si rigetta l'ipotesi nulla H_0 di presenza radice unitaria e quindi di non stazionarietà (si veda Figura 4). Una volta accertata la stazionarietà, si è proceduto ad identificare il modello individuando le componenti *AR* e *MA* sia stagionali che non stagionali (la componente *I* sia stagionale che non è già stata individuata con le differenziazioni). Per far ciò si sono nuovamente analizzati i grafici *ACF* e *PACF* ottenuti dalla *time series* (si veda Figura 3) che, come si è detto nel Capitolo 2, hanno permesso di rilevare la stagionalità settimanale. Osservando tali plot, si evidenziano infatti due discese geometriche, rispettivamente una ogni 7 ritardi nel grafico *ACF* e l'altra ogni 7 *lag* nel grafico *PACF*; ciò implica la presenza nel modello sia di una componente stagionale (settimanale) autoregressiva che una a media mobile. Sulla base delle considerazioni fatte in precedenza, il primo modello cioè il *vanilla model* che si ottiene è $ARIMA(0, 1, 0)(1, 1, 1)_7$ (modello *SARIMA*). A questo punto, per decretare se il sistema predittivo ottenuto possa essere migliorato aggiungendo anche altre componenti, si sono realizzati i plot *ACF* e *PACF* dei residui (si veda Figura 5). Osservando i grafici però, non si riesce a capire chiaramente se esistono delle componenti *AR* e *MA* non stagionali. Per questo motivo si è deciso di effettuare una *grid search* facendo variare i coefficienti *AR* e *MA* da 0 a 6 considerando, tra i vari

Value of test-statistic is: -21.74

Critical values for test statistics

	1pct	5pct	10pct
tau2	-3.43	-2.86	-2.57

Figura 4: Test Augmented Dickey-Fuller

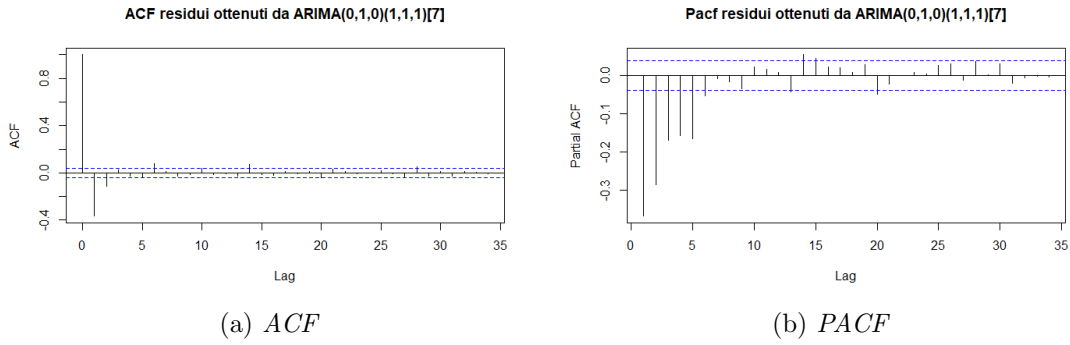


Figura 5: ACF e PACF primo modello

modelli generati, il più performante in relazione al *MAPE* registrato sul *validation set*. Alla fine del processo si ottiene così il secondo modello, migliorativo del *vanilla model*, che è $ARIMA(5, 1, 4)(1, 1, 1)_7$. Una volta implementato il secondo sistema predittivo, si è realizzato il grafico *ACF* dei residui per capire se tutta la dipendenza lineare nei dati fosse stata catturata e quindi se i residui risultassero una *white noise*. Osservando il plot (si veda Figura 6), emerge che non tutta la dipendenza lineare è stata colta in quanto meno del 95% delle bande di confidenza è interna all'intervallo sicché non si può dire che i residui formano una *white noise*. Il motivo per cui si verifica questa situazione può essere legata al fatto che nella serie temporale è presente anche una stagionalità annuale che però *ARIMA* non riesce a catturare direttamente con le componenti definite. Per questo motivo si è deciso di aggiungere al modello migliore ottenuto finora dei regressori esterni, nello specifico delle sinusoidi per catturare la stagionalità mancante; anche qui si adotta una *grid search* sul *validation set* con valori da 2 a 30 per determinare il numero ottimo di sinusoidi che risulta essere 26. Considerando le performance ottenute sul *test set* (si veda Tabella 1), emerge che l'ultimo modello realizzato risulta essere il migliore rispetto a quello ottimo ricavato finora in quanto il *MAPE* passa da 14.71% a 10.47%. Una volta individuato il nuovo sistema predittivo ottimo, si è provato ad aggiungere a tale modello altri regressori, questa volta delle variabili *dummy* che considerano le

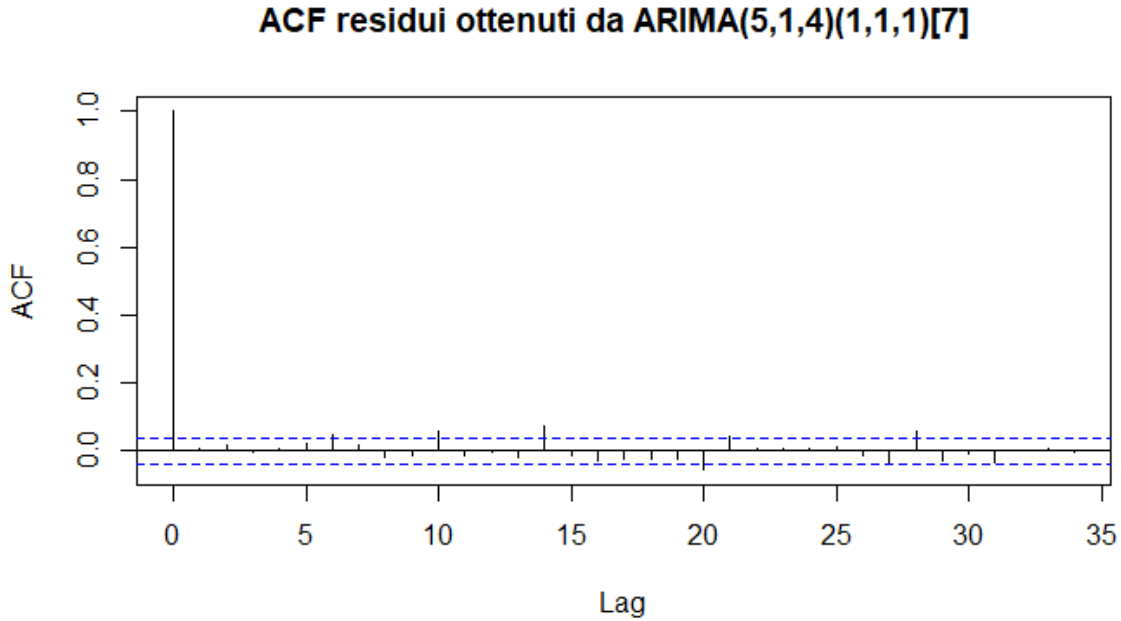


Figura 6: ACF residui secondo modello

più importanti festività celebrate in tutto il mondo come Natale e Pasqua (per l'elenco completo si veda Tabella 2). Si è effettuata questa operazione perchè si pensa che i prezzi del mercato energetico possano essere influenzati dalle festività in seguito ad un fabbisogno energetico diverso in quei giorni particolari. Valutando le performance sul *test set* (si veda Tabella 1), non si evince però alcun miglioramento rispetto al modello ottimo ricavato finora in quanto il *MAPE* non diminuisce ma aumenta perciò il sistema predittivo migliore rimane quello individuato precedentemente. A questo punto, per capire se il modello ottimo abbia o meno colto tutta la dipendenza nei dati, si è realizzato il grafico *ACF* dei residui. Osservando il plot (si veda Figura 7), si evidenzia un numero leggermente ridotto di bande di confidenza esterne all'intervallo rispetto al grafico precedente; ciò implica che un'altra parte di varianza nei dati è stata catturata anche se non tutta quella mancante visto che la quantità totale di *spike* interni all'intervallo è ancora inferiore al 95%. Questa situazione non sorprende visto che è raro identificare un modello che catturi completamente la varianza nei dati per di più avendo poche informazioni sulla loro provenienza. Si pensa quindi che il modello migliore ricavato, risulti essere il massimo che è possibile ottenere per modellare i dati a disposizione.

Tabella 1: Performance ARIMA ottenuti dal modello base sul test set

MODELLO									MAPE
ARIMA							regressori esterni		
p	d	q	P	D	Q	$stag.$	$sinusoidi$ $(stag.annuale)$	$dummy$ $(festività)$	
5	1	4	1	1	1	7	×	×	14.71%
5	1	4	1	1	1	7	✓	×	10.47%
5	1	4	1	1	1	7	✓	✓	11.18%

Tabella 2: Festività considerate

Festività
Pasqua, Pasquetta, vigilia di Natale, Natale, Santo Stefano, Capodanno, Epifania, Pentecoste

ACF residui ottenuti da ARIMA(5,1,4)(1,1,1)[7] + reg (stag.)

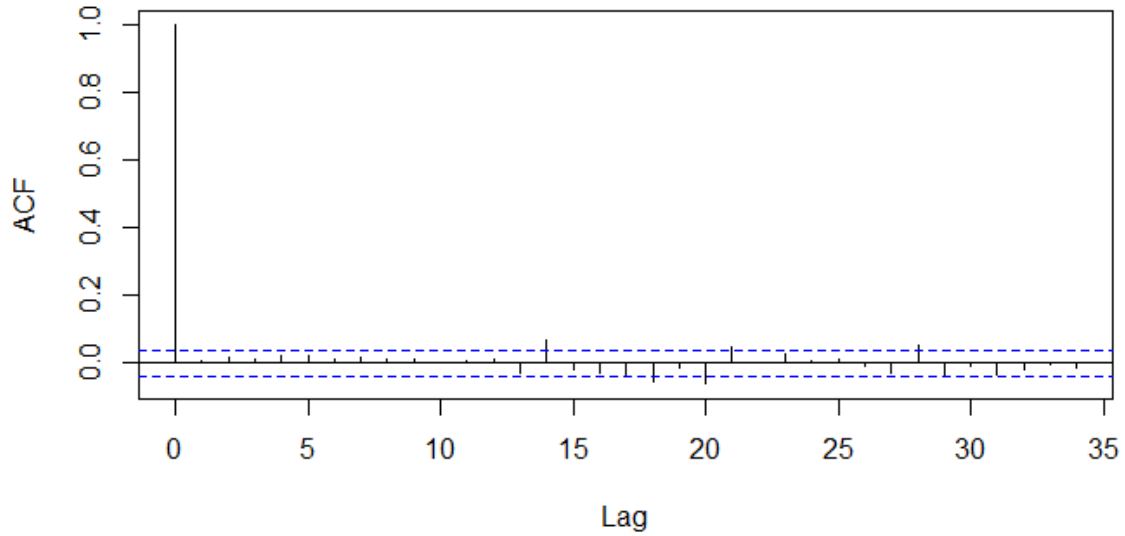


Figura 7: ACF residui ARIMA migliore

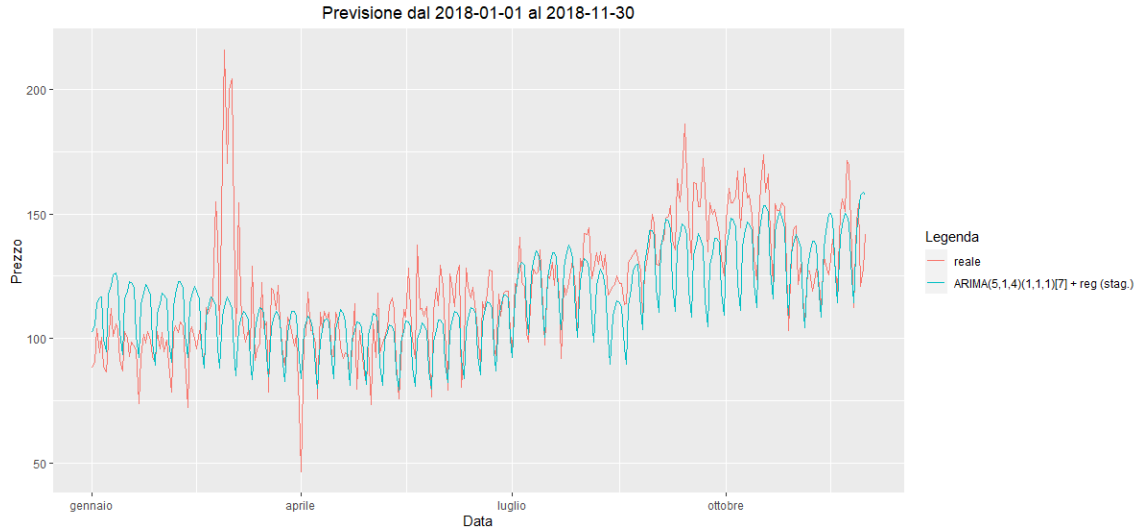


Figura 8: Previsione ARIMA migliore

5.2 UCM

Il secondo sistema predittivo è *UCM* (acronimo di *Unobserved Components Model*), un modello che decompone la serie temporale in 4 elementi: *trend*, *ciclo*, *stagionalità* e *white noise*. Il primo elemento è responsabile delle variazioni della media del processo nel lungo periodo, il secondo rappresenta un fenomeno ciclico, il terzo si occupa della deviazione del livello medio della serie storica in alcuni istanti temporali infine il quarto elemento è attribuibile all'errore con cui si misurano le serie storiche [5]. Queste componenti, ad eccezione del *white noise*, vengono tutte derivate da funzioni deterministiche del tempo come retta e sinusoidi, rese stocastiche attraverso degli opportuni shock casuali.

5.2.1 Procedimento

Così come si è visto per *ARIMA*, anche qui si realizza il modello *UCM* procedendo per step migliorativi. Tutti i sistemi implementati nelle varie fasi, sono stati rappresentati secondo la forma *state space* e la stima dei parametri è avvenuta utilizzando il filtro di *Kalman*. La prima operazione effettuata è stata determinare il *vanilla model*, definito da due componenti quali *trend* e *stagionalità* (settimanale); il primo elemento viene modellato con un *random walk* (*RW*) mentre il secondo con delle *dummy* stocastiche. Si è deciso di modellare la stagionalità settimanale con delle *dummy* e non con regressori trigonometrici perché, attraverso prove empiriche, non si riscontrano performance

differenti sul *test set* tra le due strategie e quindi, si è deciso di optare per la soluzione computazionalmente meno pesante cioè le *dummy*. Una volta definito il *vanilla model*, si è avviata la fase di *refine* di tale modello includendo innanzitutto dei regressori trigonometrici utili a modellare l'altra stagionalità presente nei dati ovvero quell'annuale. Si è scelto di utilizzare questa tipologia di covariate perchè da letteratura, la trigonometria risulta essere la migliore soluzione per modellare questo tipo di stagionalità. Per decretare il numero ottimo di sinusoidi, si è effettuata una *grid search* facendo variare l'iperparametro da 2 a 30. Alla fine del processo si ottiene 19 come numero ottimo di sinusoidi le quali, incluse nel primo modello, consentono a tale sistema di raggiungere sul *test set* un *MAPE* del 15.80%. Una volta modellate le due stagionalità presenti nei dati, si è implementato un nuovo sistema predittivo che, a differenza di quello definito precedentemente, cattura il trend con un *local linear trend (LLT)*. Confrontando le performance sul *test set* (si veda Tabella 3), non si evidenzia però nessun miglioramento rispetto al modello ottimo ovvero il *vanilla model* migliorativo. A questo punto, si è provato come ultima operazione ad aggiungere al sistema predittivo migliore, sulla base delle considerazioni fatte per *ARIMA*, le variabili *dummy* che considerano le festività. Confrontando le performance sul *test set* (si veda Tabella 3), l'aggiunta di tali regressori contribuisce a ridurre ulteriormente il *MAPE* sicchè il sistema predittivo ottimo risulta essere l'ultimo modello che considera anche le festività. Per quanto riguarda il sistema predittivo migliore, si è deciso di non includere alcuna componente ciclica in quanto non si hanno informazioni sufficienti per decretare se nel mercato energetico di riferimento, esista o meno un ciclo economico o di altra natura.

Tabella 3: Performance UCM ottenuti dal modello base sul test set

<i>trend</i>	MODELLO			MAPE
	<i>stag. settimanale</i>	<i>stag. annuale</i>	<i>regressori esterni (festività)</i>	
RW	dummy	reg. trigonometrici	×	15.80%
LLT	dummy	reg. trigonometrici	×	16.93%
RW	dummy	reg. trigonometrici	✓	11.47%

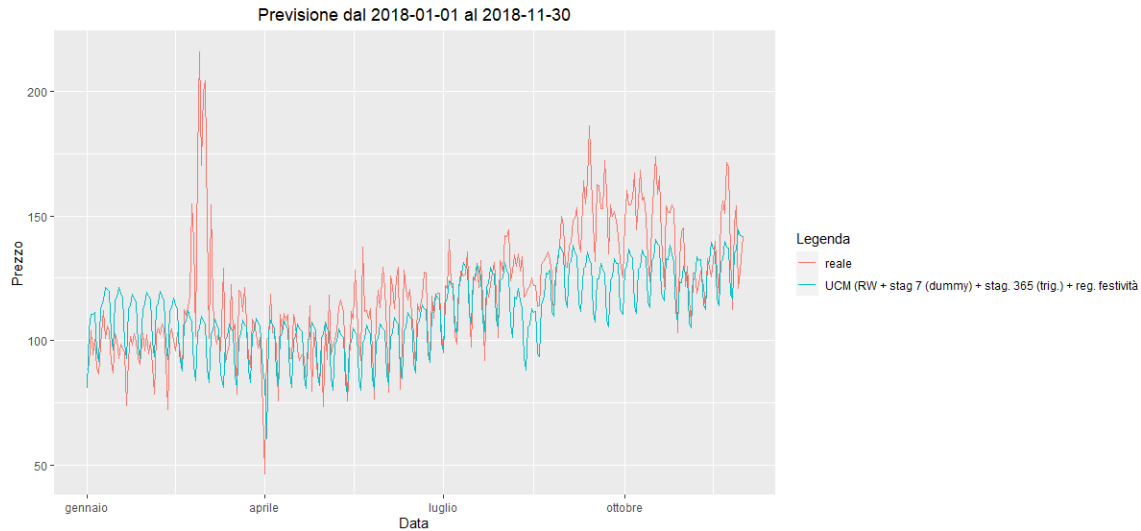


Figura 9: Previsione UCM migliore

5.3 RNN

Il terzo modello realizzato è un tipo particolare di rete neurale che si chiama *RNN* (acronimo di *Recurrent Neural Network*). Questa rete, considerando l'architettura generale, prevede che ad ogni *hidden layer* non venga passato solo il valore X_t come invece avviene per una classica *feed forward*, ma anche l'output dello strato nascosto al tempo $t - 1$; ciò implica che il risultato ottenuto all'istante temporale t non dipende solo da X_t ma anche da tutte le elaborazioni effettuate fino all'istante precedente [6]. In questo modo all'interno della rete neurale si ottiene una memoria, rappresentata dallo stato h_t , che tiene traccia di tutte le elaborazioni fatte in precedenza catturando così la sequenzialità tra input e output. Esistono varie architetture che si possono adottare per realizzare una *RNN*, le più utilizzate sono *LSTM* e *GRU* le quali, attraverso il meccanismo dei *gate*, riescono a modellare sia le dipendenze a lungo termine che quelle a breve termine.

5.3.1 Procedimento

Per realizzare la rete neurale si è dovuto prima svolgere un opportuno preprocessing in modo tale che il modello possa utilizzare in maniera adeguata i dati a disposizione. Le principali attività svolte sono state la standardizzazione dei dati e la creazione del tensore. La prima operazione è stata effettuata per rendere più performante la rete

neurale, nello specifico si è applicata la seguente formula a tutti i dati a disposizione:

$$z = \frac{x - \mu}{\sigma} \quad (2)$$

dove μ e σ sono rispettivamente la media e la deviazione standard calcolata sui dati di training.

La seconda operazione invece, è stata effettuata perchè la rete neurale richiede come input un tensore con 3 *axes* così definiti: il primo indica il numero di esempi di training, il secondo rappresenta la *window size*, ovvero quanti istanti temporali (nel caso di riferimento sono i giorni visto che i dati sono giornalieri) deve utilizzare il modello per predire n istanti successivi infine il terzo indica il numero delle feature utilizzate per effettuare le previsioni. Considerando la *window size*, si è scelto come numero 365 visto che, dopo diverse prove empiriche effettuate con il *vanilla model* (*RNN* di tipo *LSTM* senza alcun *hidden layer*), si è rilevata essere la migliore finestra temporale per allenare la rete neurale. Per quanto riguarda l'architettura del *vanilla model*, si è impostato come numero di neuroni nello strato di input un valore pari a 100 (valore arbitrario) mentre per l'*output layer* si è settato un valore pari a 334, dato l'orizzonte temporale di previsione futuro definito nel progetto. Una volta implementato il modello base, si è avviata la fase di *refine* di tale sistema eseguendo una serie di operazioni. La prima attività effettuata è stata ottimizzare l'architettura, nello specifico trovare il numero ottimo di neuroni nello strato di input tramite *AutoML* lasciando invariata l'ampiezza dell'*output layer*. Oltre a questo, per evitare il fenomeno dell'*overfitting*, si sono applicate anche due tecniche di regolarizzazione ovvero *Early Stopping* e *l2 regularizers*. Per quanto riguarda la prima tecnica, si è impostata come metrica da monitorare la *loss* della validation e come *patience* (numero di epoche entro le quali la *loss* non peggiori) un valore pari a 10. Considerando la seconda tecnica invece, applicata solo sull'*input layer* contenente le celle *LSTM*, la *weight regularization* è stata definita in maniera automatica attraverso *AutoML*. Una volta definita la nuova rete neurale, migliorativa del *vanilla model*, si è provato a migliorarla implementando due nuove architetture che prevedono l'aggiunta di uno e due *hidden layer Dense* rispetto alla struttura iniziale. Visto l'aumento della complessità dell'architettura e quindi il rischio maggiore di *overfitting*, si è deciso di applicare ai due nuovi modelli, oltre alle tecniche di regolarizzazione viste precedentemente, anche *Dropout* ai layer aggiunti ottimizzando *dropout rate* sempre tramite *AutoML*. Tutti i modelli, compreso quello iniziale, sono stati addestrati utilizzando dei *batch* di dimensione 128 e settando il numero massimo di epoche a 100. Considerando la *loss* e l'ottimizzatore, per i tre modelli si è optato per *MSE* e *RMSProp*. Per quanto riguarda la *loss* la scelta è ricaduta su *MSE* in quanto, da letteratura, risulta essere una delle migliori misure da utilizzare quando si vuole addestrare una rete neurale per effettuare delle previsioni. Considerando invece

l'ottimizzatore, si è scelto *RMSProp* perché, rispetto agli altri candidati tra i quali *Adam*, si rivela quello da cui si ottengono le performance più alte. Per quanto riguarda le funzioni di attivazione, la scelta è ricaduta sulla *tanh* per l'*input layer*, *relu* per gli strati nascosti e *linear* per *output layer*. La prima è stata utilizzata perché, da letteratura, risulta essere particolarmente adatta quando il layer contiene celle *LSTM*, la seconda perché tra tutte le funzioni è quella che velocizza maggiormente il training dei pesi infine la terza è stata impiegata perché la rete viene utilizzata per un problema di regressione di conseguenza l'output deve essere un valore reale numerico e non una probabilità. Per decretare il numero di neuroni contenuti in ogni layer ad eccezione dello strato di output, si è utilizzato, come si è visto per il *vanilla model*, *AutoML* che a sua volta è stato impiegato anche per determinare il valore ottimo di *learning rate* del *RMSProp*. Osservando i risultati ottenuti sul test set (si veda Tabella 4), emerge che l'ultima architettura *deep* ottenuta, ovvero quella costituita da 4 layer di cui 2 nascosti, risulta essere quella da cui si ottengono le performance più alte.

Tabella 4: Performance RNN ottenuti dal modello base sul test set

MODELLO							MAPE
<i>input layer</i> [2,1024]	<i>hidden layer 1</i> [2,1024]	<i>hidden layer 2</i> [2,1024]	<i>output layer</i>	<i>dropout rate</i> [0.05,0.4]	<i>learning rate</i> [1E ⁻⁴ ,1E ⁻³]	<i>l2 reg.</i> [1E ⁻⁶ ,1E ⁻³]	
898	×	×	334	×	9.95E ⁻⁴	1.25E ⁻⁵	23.11%
956	170	×	334	0.24	7.79E ⁻⁴	2.44E ⁻⁴	22.28%
364	737	954	334	0.24	9.50E ⁻⁴	3.99E ⁻⁴	21.62%

A questo punto, visto che le performance ottenute con il modello migliore non sono per nulla paragonabili a quelle ricavate con *ARIMA* e *UCM*, si è provato a modificare il modo in cui venivano svolte le previsioni applicando questa volta il metodo ricorsivo (previsione 1 passo in avanti). Per far ciò si è deciso di utilizzare la stessa architettura (2 *hidden layer*) così come gli stessi iperparametri ottimi ricavati precedentemente dal processo *AutoML* con l'unica differenza riguardante l'ampiezza dell'*output layer* settato a 1. Osservando i risultati ottenuti sul *test set* (si veda Tabella 5), si evidenzia un concreto miglioramento con la nuova strategia infatti il *MAPE* registrato passa da 21.62% a 12.46%. Questa diversità delle performance, può essere legata al fatto che la rete neurale con una previsione diretta, non riesce a catturare i pattern all'interno della serie temporale mentre effettuando una previsione 1 passo in avanti, riesce gradualmente a catturarli visto che è in grado di predire con una buona accuratezza, il giorno successivo avendo a disposizione i 365 giorni passati. Per questo motivo l'ultima strategia si

rivela più efficace dell'altra sicché il modello considerato ottimo risulta essere la *RNN* costituita da 4 layer di cui 2 nascosti e con un solo neurone nello strato di output.

Tabella 5: Performance RNN migliore sul test set utilizzando le due strategie

MODELLO	MAPE	
	<i>prev. diretta</i>	<i>prev. 1 passo in avanti</i>
RNN (2 hidden layer)	21.62%	12.46%

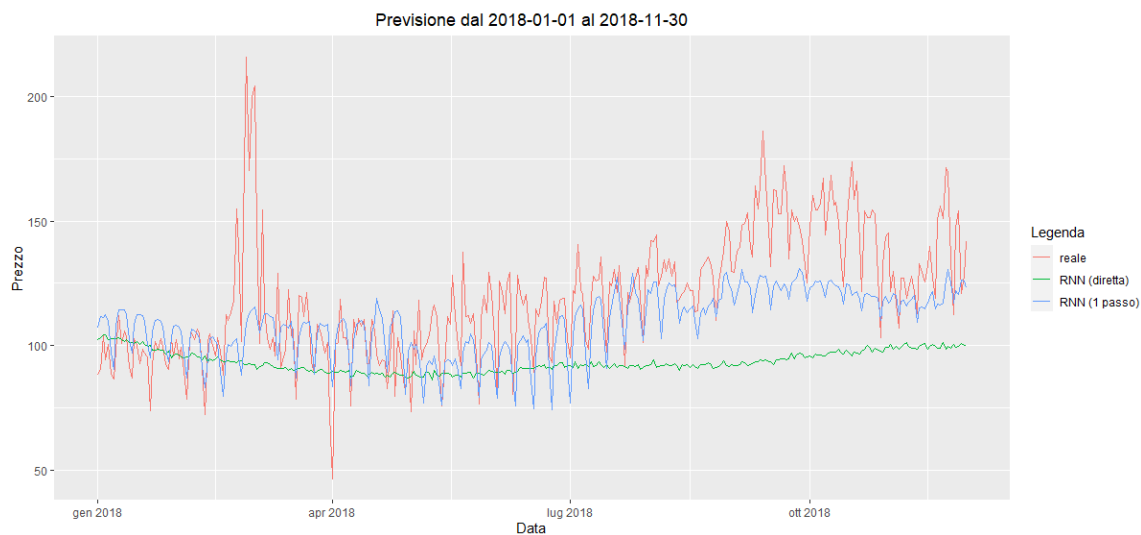


Figura 10: Previsione RNN migliore utilizzando le due strategie

6 Risultati

Una volta individuato i modelli migliori per ciascuna delle strategie implementate, si è effettuato un loro confronto per valutarne complessivamente le performance. A tal proposito si sono considerate per tutti i modelli le previsioni "one shot" in modo tale che la comparazione avvenga sulla base di uno stesso criterio; per questo motivo la rete neurale confrontata con gli altri sistemi non risulta essere quella da cui ottengono le

performance più alte, ovvero quella che effettua le previsioni ricorsive bensì la *RNN* ottima che svolge previsioni dirette (ampiezza *output layer* pari a 334). Osservando la tabella riassuntiva dei risultati (si veda Tabella 6), si possono trarre una serie di considerazioni; la prima è che il modello ottimo, considerando unicamente le performance del *test set*, risulta essere *ARIMA* a seguire poi *UCM*. La rete neurale, diversamente dalle aspettative, non si rivela il modello migliore. Questo può essere legato da due fattori; il primo è che forse la *RNN*, per catturare meglio la dipendenza nei dati, necessita di un tensore particolare in cui devono essere aggiunte delle informazioni implicite che aiutano a catturare maggiormente i pattern all'interno della serie temporale. Il secondo fattore invece può essere legato all'architettura intrinseca della rete troppo semplice di conseguenza, per ottenere delle performance alte, si dovrebbe per esempio aumentare in maniera consistente il numero di strati nascosti. Un'altra considerazione che si può fare osservando la tabella riassuntiva è che i primi due modelli tendono ad adattarsi troppo ai dati di training infatti sia per *ARIMA* che *UCM* si riscontra un problema di *overfitting*, impercettibile nel primo caso ben accentuato nell'altro. L'ultimo modello invece, non riscontra tale problema però dall'altro lato le performance ottenute sia sul *training set* che sul *test set*, risultano essere nettamente inferiori rispetto agli altri due sistemi predittivi. Per questo motivo, anche se la rete neurale non è soggetta al problema descritto in precedenza, si predilige comunque un sistema predittivo in cui esiste un leggero *overfitting* ma che in fase di test si rivela il più performante. A tal ragione, si decreta quindi che il modello migliore tra i tre implementati risulta essere *ARIMA*.

Tabella 6: Performance modelli migliori

MODELLO	MAPE	
	<i>training set</i>	<i>test set</i>
ARIMA	9.68%	10.47%
UCM	7.50%	15.80%
RNN	23.84%	21.62%

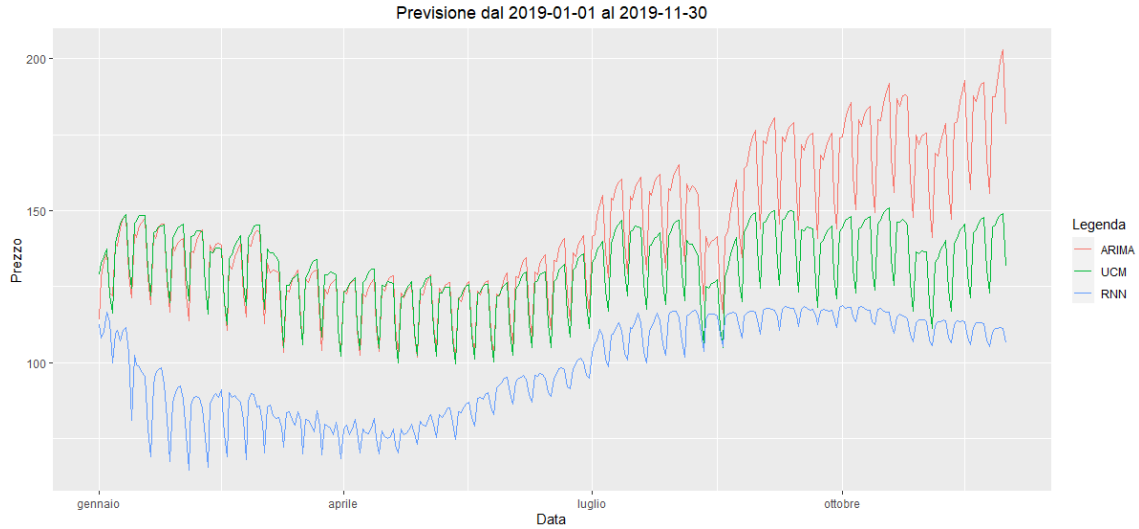


Figura 11: Previsioni finali modelli migliori

7 Conclusioni

Nel progetto si sono presentate tre diverse strategie per realizzare un modello che possa prevedere, con un certo livello di accuratezza, i prezzi giornalieri di un mercato energetico in 11 mesi. I risultati ottenuti mostrano che il miglior sistema predittivo, diversamente da quanto si poteva aspettare, non risulta essere uno di *Machine Learning* bensì uno statistico ovvero *ARIMA*. Il merito di questo "successo" può essere legato fondamentalmente all'aggiunta a tale modello di regressori esterni, nello specifico sinusoidi che, oltre ad essere utili per migliorare il modello *UCM*, aiutano *ARIMA* a catturare la stagionalità annuale, pattern che dovrebbe essere colto automaticamente dalla rete neurale ma che in realtà non riesce a catturare nè con una previsione diretta nè con una ricorsiva. Per quanto riguarda invece gli altri regressori che considerano le festività, si riscontrano esiti differenti tra i due modelli in cui sono stati inclusi, nello specifico per *ARIMA* i regressori non contribuiscono ad un miglioramento complessivo delle performance mentre per *UCM* vale la situazione opposta. A fronte dei risultati riscontrati, si registrano due possibili miglioramenti futuri da apportare al progetto. Il primo sviluppo riguarda l'utilizzo di modelli diversi di *Machine Learning* (es. *SVR*) e verificare se le performance ottenute risultino essere migliori di quelle sia della rete neurale che del modello *ARIMA*. Il secondo sviluppo invece, consiste nel cambiare completamente l'architettura della rete neurale definita e verificare anche qui se le modifiche effettuate, portino o meno a dei miglioramenti evidenti.

Riferimenti bibliografici

- [1] C. Chatfield, *Time-series forecasting*. CRC press, 2000.
- [2] B. Isabelle, “Machine learning: una rivoluzione nella supply chain.” [Online]. Available: <https://www.generixgroup.com/it/blog/machine-learning-una-rivoluzione-nella-supply-chain>
- [3] Sas, “Analisi predittiva - che cos'è e perché è importante.” [Online]. Available: https://www.sas.com/it_it/insights/analytics/predictive-analytics.html
- [4] J. Brownlee, “How to create an arima model for time series forecasting in python.” [Online]. Available: <https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/>
- [5] M. M. Pelagatti, “Previsioni aziendali e valutazione dinamica dell'impatto di campagne pubblicitarie, eventi esterni e investimenti in marketing,” *Università degli Studi di Milano-Bicocca per SAS Institute*, p. 72.
- [6] A. Provino, “Cos'è una recurrent neural network — rnn.” [Online]. Available: <https://andreaprovino.it/rnn-recurrent-neural-network/>