Progetto di Streaming Data Management And Time Series Analysis

Davide Sangalli - Matricola: 848013

January 10, 2020

Abstract

In questo progetto si vuole analizzare l'andamento dei prezzi del mercato energetico, aggregati per giorno, e fare previsione sui futuri prezzi per il periodo *1/1/2019-30/11/2019*.

1 Data Manipulation

Il dataset è stato diviso in train e validation set, nel seguente modo:

- *train_set*: dall'1/1/2010 al 31/12/2017
- *validation_set*: dall'1/1/2018 al 31/12/2018

2 Modello SARIMA

Il primo modello sviluppato è un modello lineare di tipo SARIMA.

2.1 Test di stazionarietà

E' stato innanzitutto effettuato un test di stazionarietà (in senso debole), controllando se la media e la varianza della serie originale fossero costanti nel tempo. A questo scopo, sono state definite le funzioni *check_variance_stationariety* e *check_mean_stationariery*.
Da questi test è risultato che la serie di partenza non era stazionaria, perciò si è operata la trasformazione logaritmica, che l'ha resa quasi stazionaria.
Partendo dunque dalla serie logaritmica, si è sviluppato il modello SARIMA.

2.2 Stima del modello

Osservando i grafici di ACF e PACF, si è provato un modello con 2 lags non stagionali e una differenza stagionale di periodo 7. Il modello risultante era buono, ma, osservando i grafici ACF e PACF dei residui, si è deciso di aggiungere un ordine di *moving average* sia alla componente stagionale che a quella non stagionale. Grazie a quest'ultimo accorgimento, il modello sembrava cogliere l'autocorrelazione rimasta tra i residui. Il modello SARIMA risultante è perciò il seguente (l'ordine delle componenti è AR,I,MA):

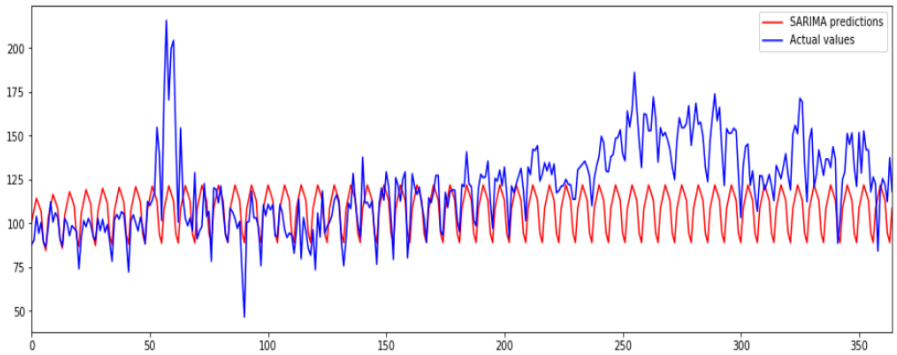
- componente non stagionale: *(2,0,1)*
- componente stagionale: *(0,1,1)*, con stagionalità 7

2.3 Previsoni

Usando il modello stimato precedentemente, sono state effettuate delle previsioni sia sul *train_set* che sul *validation_set*.
In particolare, le previsioni sul *validation_set* sono state effettuate in questo modo:

- è stata inizializzata una *history* per il modello, che inizialmente contiene tutti e soli i valori del train set
- si esegue un ciclo su tutta la lunghezza del validation set e, ad ogni iterazione, viene calcolata una previsione, che viene successivamente aggiunta alla *history* del modello (perciò, all'*iesima* iterazione, la history sarà composta dai valori di train più i previsioni)
- si ripete il punto precedente per tutti gli elementi all'interno del validation set

In questo modo si riesce a comparare la bontà previsiva del modello rispetto ai veri valori del validation set.
Di seguito è riportato il grafico di tale comparazione.



Si nota come le previsioni riescano a seguire più o meno bene l'andamento della serie, anche se non riescono a coglierne i picchi.

3 Modello UCM

Il secondo modello proposto è un modello a componenti non osservabili.

3.1 Stima del modello

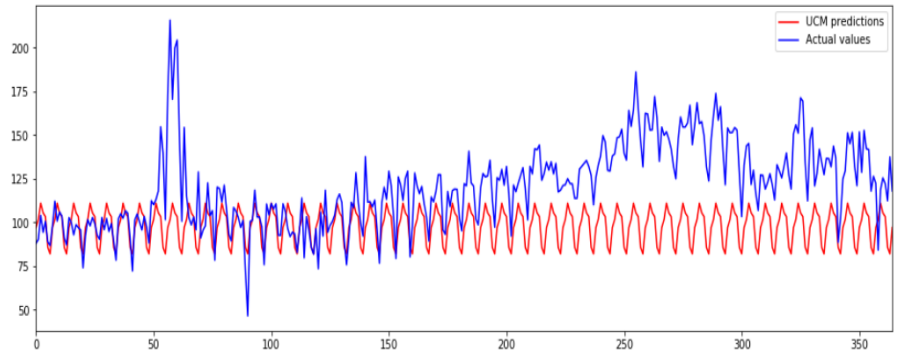
Il modello UCM che è stato costruito ha i seguenti parametri:

- *level:'random walk'*
- *seasonal: 7*

Il parametro *level* modella la componente di trend, settandola come un *random walk*, mentre il parametro *seasonal* modella la componente stagionale, sfruttando le *dummy stagionali*.
La scelta di questi parametri è avvenuta dopo aver provato diverse configurazioni delle componenti stagionali e di trend. Quello presentato è il setting grazie al quale il modello UCM raggiunge le performance migliori in termini di previsioni sul validation set.
Si è provato ad aggiungere anche una componente di ciclo stocastico, ma questo non portava a miglioramenti effettivi del modello, perciò si è deciso di ometterla nella versione finale.

3.2 Previsioni

Usando il modello UCM descritto precedentemente, sono state effettuate delle previsioni su *train e validation set*.
Il ragionamento seguito per ottenere le previsioni sul validation set è del tutto analogo a quello fatto per il modello SARIMA precedente.
Viene di seguito presentato il grafico comparativo.



Come si può notare, le previsioni di questo modello riescono solo in parte a catturare il vero andamento della serie storica, non riuscendo a seguirne i picchi. Si può intuire che il modello UCM ha delle performance leggermente peggiori nelle previsioni sul validation set rispetto al modello SARIMA.

4 Recurrent Neural Network

L'ultimo modello proposto è un modello non lineare, preso dal mondo del Machine Learning; in particolare è stata utilizzata una rete neurale ricorrente.

4.1 Costruzione del modello

E' stato costruito un modello di rete neurale ricorrente strutturato in questo modo:

- una *input_shape=(1,look_back)*, dove *look_back* è stato scelto pari a 365
- un layer *GRU (gated recurrent unit)*, con 128 neuroni
- un layer di *Dropout* con *dropout_rate* di 0.2
- un layer *Dense* con un solo neurone, per poter restituire la previsione
- il modello è stato compilato utilizzando come funzione di perdita il *mean_squared_error* e come ottimizzatore *adam*
- il modello è stato infine allenato per 15 epoche

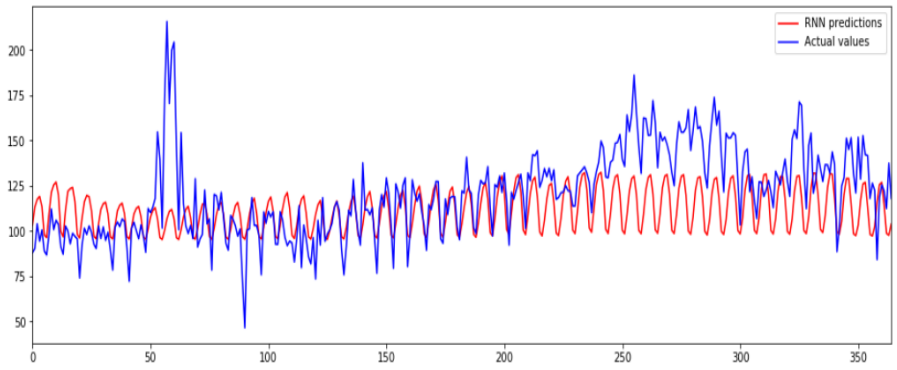
4.2 Creazione della sequenza di training

Per questo modello sono state create delle sequenze di training, ognuna delle quali è strutturata nel seguente modo:

- ogni sequenza è composta da 365 valori (corrispondenti a 365 giorni) più un valore di target, corrispondente al 366esimo giorno. Sostanzialmente si usano 365 "ritardi" per questo tipo di modello.
- le sequenze si differenziano l'una dall'altra per uno shift unitario. (i.e. la prima sequenza va dall' 1/1/2010 al 31/12/2010 e il giorno di target per questa sequenza è l' 1/1/2011; la seconda sequenza va dal 2/1/2010 al 1/1/2011 e il giorno di target per quest'altra sequenza è il 2/1/2011, ecc...).
- si è scelto 365 come numero di "ritardi" per la sequenza perchè, dopo diverse prove, si è notato che questo è il valore per il quale si hanno previsioni migliori sul validation set.

4.3 Previsioni

Usando la rete neurale ricorrente descritta precedentemente, sono state effettuate delle previsioni su *train e validation set*. Viene qui riportato solo il grafico relativo al *validation_set*.



Si nota come, anche in questo caso, le previsioni riescano a cogliere l'andamento generale del modello, senza però riuscire a seguirne i picchi. Tuttavia, le previsioni di questo modello risultano più accurate rispetto a quelle effettuate usando i due modelli precedenti.

5 Confronto tra modelli e conclusioni

La misura di errore scelta è l'*RMSE*. Si è scelta questa misura, poichè tramite l'RMSE i grossi errori commessi sui picchi di consumo assumono un peso maggiore.
Di seguito viene riportata una tabella comparativa dei tre modelli, in cui, per ciascuno, viene riportato l'errore sul train set e sul validation set.

Model	Train Error	Validation Error
SARIMA	0.176242	0.211045
UCM	0.215026	0.270935
GRU	0.291678	0.190428

Per quanto riguarda il validation set, si nota che SARIMA e GRU hanno una performance simile (è leggermente meglio GRU), mentre UCM è il peggiore dei tre, perchè ha un valore di RMSE più alto. Sul train set, si comporta meglio SARIMA, seguito da UCM ed infine da GRU.

NB: per fare le previsioni dall' 1/1/2019 al 30/11/2019, si sono allenati di nuovo tutti e tre i modelli, usando tutto il dataset a disposizione. Le relative previsioni sono poi state salvate in un file csv.