Streaming Data Management and Time Series Analysis
Report del progetto

Edoardo Gervasoni 790544 - e.gervasoni4@campus.unimib.it

Università di Milano-Bicocca — July 7, 2020

Introduzione

L'obiettivo di questo progetto è quello di definire, sviluppare e validare un sistema predittivo per dati di tipo time series scaturiti da un'aggregazione dei prezzi del mercato energetico. Ciò è stato fatto grazie all'implementazione di tre diverse tipologie di modelli: ARIMA, UCM e Machine Learning. In particolare, i migliori modelli per ciascuna categoria sono stati impiegati per compiere previsioni riguardo a valori che vanno dal 1 gennaio 2019 al 30 Novembre 2019. Di seguito presentiamo una descrizione di quanto è stato fatto.

1 Dati

I dati in nostro possesso sono una serie storica giornaliera, che va dal 1 Gennaio 2010 al 31 Dicembre 2018. Più nello specifico, il file a disposizione è composto da due colonne, **Data** e **Value** (il prezzo), e da 3287 osservazioni.

Data <date>	value <dbl>
2010-01-01	41.65104
2010-01-02	131.28660
2010-01-03	117.38812
2010-01-04	116.46128
2010-01-05	123.82376
2010-01-06	104.28556

Figure 1: Come si presentano le prime 6 righe del dataset.

Dopo aver letto i dati, si è proceduto con la suddivisione di questi in *training set* e *validation set*. Tale partizione è stata fatta considerando per il primo tutti i dati dal 1 Gennaio 2010 al 31 Dicembre 2016, mentre al secondo sono stati assegnati i dati appartenenti agli ultimi due anni. In percentuali, circa il 78% delle osservazioni appartiene al training set, mentre il 22% al validation.

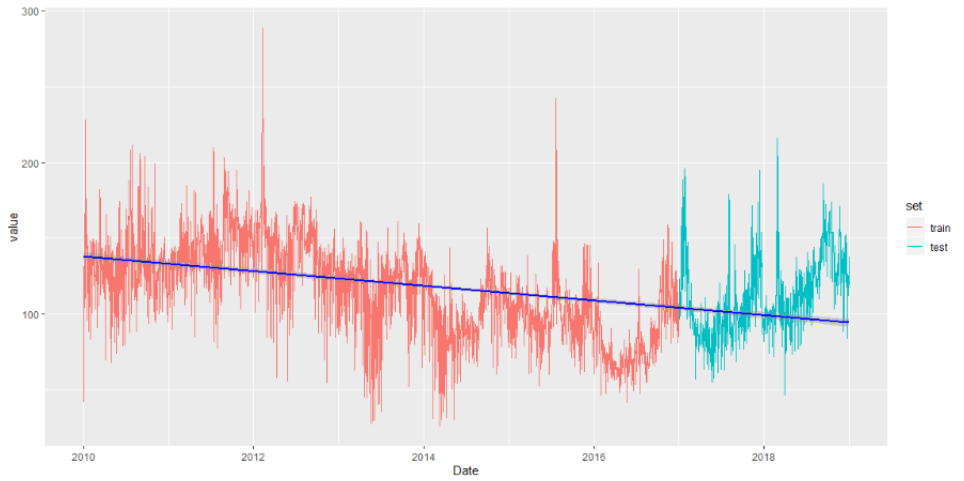

Figure 2: Suddivisione dei dati tra training set (in arancio) e validation set (in azzurro)

2 Misurazione della bontà dei modelli

Per andare a stimare le performances dei modelli sviluppati, è stato scelto di considerare il *mean absolute percentage error*, o MAPE. Questa grandezza ha formula:

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \tag{1}$$

in cui A_t è il valore vero dell'osservazione al tempo t, mentre F_t è il valore predetto sempre al tempo t. Questa grandezza è stata scelta per misurare la bontà dei modelli per la sua praticità, in quanto permette di comprendere in modo semplice e chiaro quanto i dati previsti deviano da quelli reali in termini percentuali. Un probelma dell'impiego di questa grandezza si ha quando i valori al denominatore sono prossimi a zero, nel qual caso il rapporto tende a divergere; tuttavia, dato che i nostri dati non presentano questa caratteristica, non vi sono controindicazioni all'utilizzo del MAPE.

3 ARIMA

La prima tipologia di modelli ad essere stata sviluppata è quella ARIMA, ovvero Autoregressive Integrated Moving Average. Il passo iniziale è consistito nel calcolo del parametro λ della trasformazione di Box Cox. Dato che questo è risultato essere 0.93, e dunque prossimo a 1, non è stato necessario applicare alcuna trasformazione ai dati. Quindi si è proceduto andando a visualizzare i grafici ACF e Partial ACF dei dati.

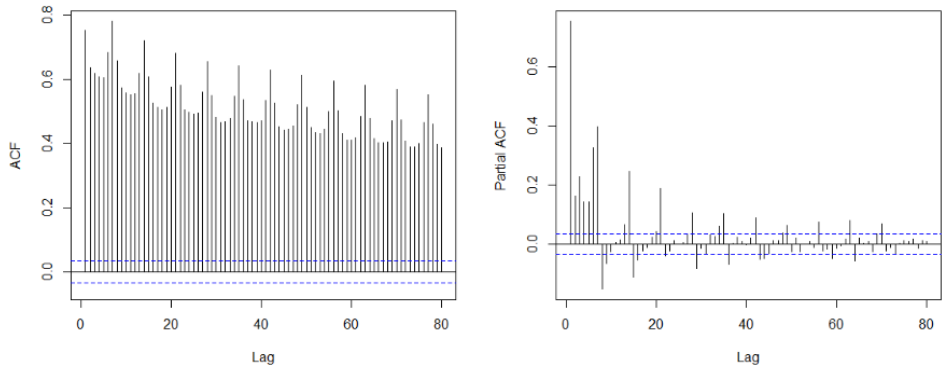

Figure 3: ACF e Partial ACF

Come si può notare dai correlogrammi in figura 3, i dati in nostro possesso presentano una stagionalità settimanale.
Si è quindi proceduto andando a considerare un modello di prova ARIMA(0,0,0)(1,1,1)[7] che tenesse conto della stagionalità di periodo 7, e quindi sono stati analizzati i correlogrammi dei residui.

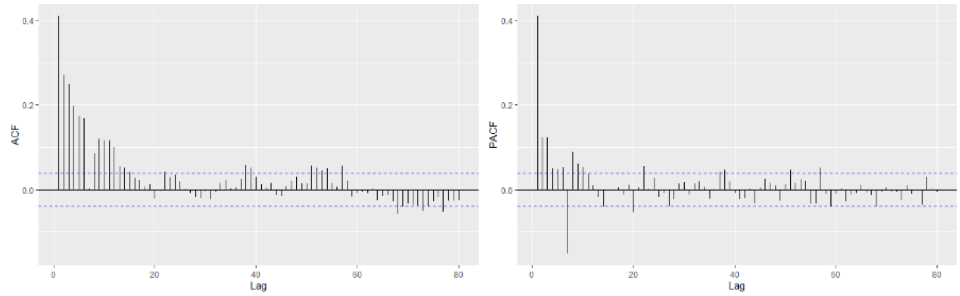

Figure 4: ACF e Partial ACF dei residui di ARIMA(0,0,0)(1,1,1)[7]

I grafici mostrano la presenza di varie componenti di cui si deve tener conto; in particolare, i valori elevati dei primi 6 lag dell'ACF potrebbero suggerire una componente Moving Average (6). Di conseguenza è stato deciso di procedere andando a considerare diversi modelli ARIMA variando le componenti AR e MA non stagionali, e vedere quali tra questi ottenessero i risultati migliori in termini di log-likelihood e AIC sul training set. Dato che l'ARIMA(6,0,7)(1,1,1)[7] è risultato essere il migliore tra tutti quelli valutati, si è stabilito di continuare la nostra trattazione con questo.

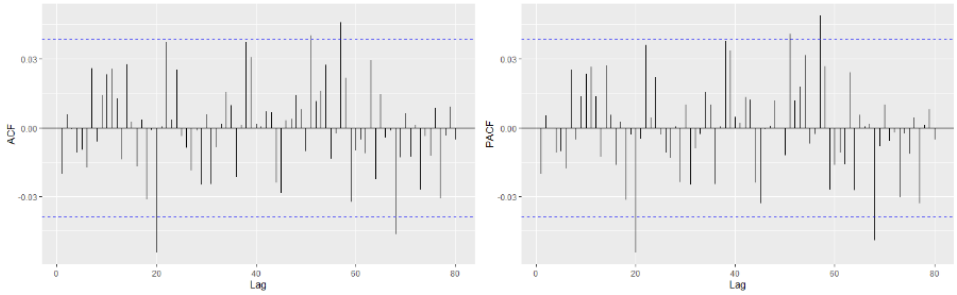

Figure 5: ACF e Partial ACF dei residui di ARIMA(6,0,7)(1,1,1)[7]

L'analisi dei correlogrammi dei residui del nuovo modello ARIMA mostra ancora la presenza di diverse compnenti da considerare; in particolare ora è il lag 20 ad essere significativamente diverso da 0.
Si è quindi proceduto con la valutazione delle radici del polinomio AR, che sono risultate essere pari a 0.6787253, 0.6762618, 0.6762618, 0.8672084, 0.6787253 e 0.6369676. Dati i loro valori lontani dall'unità, non è stata necessaria alcuna integrazione.
Successivamente, per migliorare ulteriormente il modello, si è proseguito con l'aggiunta di regressori esterni con l'obiettivo di catturare le componenti che non sono state ancora colte. Come primo passo è stata considerata l'aggiunta di 24 armoniche di stagionalità annua, e successivamente, grazie alla libreria R **RQuantLib**, si è considerata l'aggiunta di una variabile dummy *holiday* che tenesse conto delle festività italiane dal 1 Gennaio 2010 al 31 Dicembre 2018. Da una loro valutazione sul training set, il secondo è risultato migliore sia in termini di log-likelihood che in termini di MAPE.

	MAPE sul Training set
ARIMA(6,0,7)(1,1,1)[7] con armoniche	0.09375075
ARIMA(6,0,7)(1,1,1)[7] con armoniche e dummy vacanze	0.08955639

Infine sono stati stimati i risultati dei due diversi modelli sul validation set, in modo da stabilire quale dei due risultasse il più adatto per essere utilizzato per compiere predizioni sui valori dal 1 Gennaio al 30 Novembre 2019. Di seguito presentiamo quanto ottenuto:

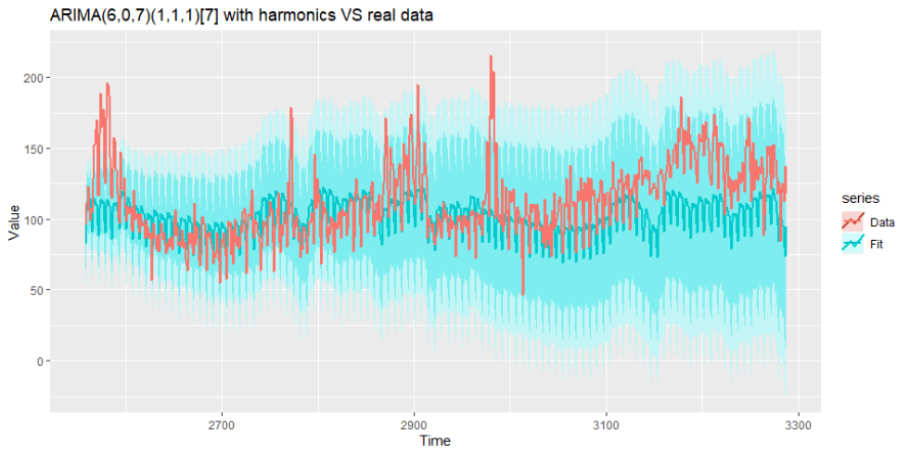

Figure 6: Confronto tra predizione del modello con armoniche e dati reali sul validation set

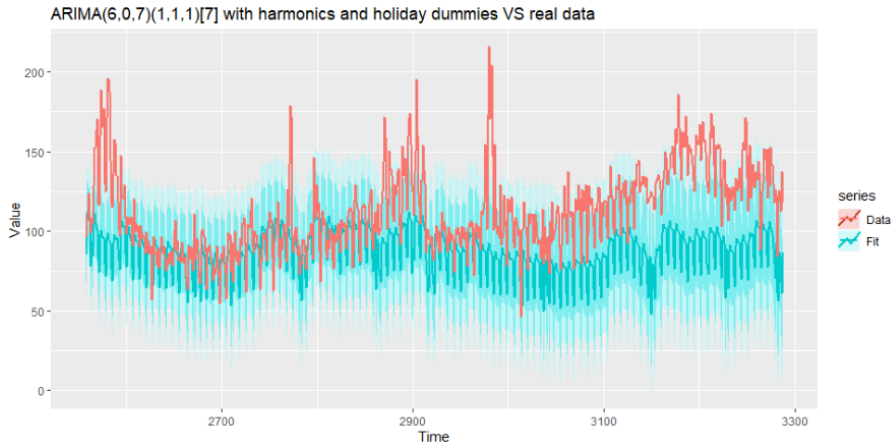

Figure 7: Confronto tra predizione del modello con armoniche e con dummy vacanze e dati reali sul validation set

MAPE ottenuto dai due diversi modelli sul validation set:

	MAPE
ARIMA(6,0,7)(1,1,1)[7] con armoniche	0.156637
ARIMA(6,0,7)(1,1,1)[7] con armoniche e dummy vacanze	0.2202016

Dai grafici e dai risultati in termini di MAPE, possiamo vedere che il modello con le sole armoniche è quello che ha ottenuto performances migliori, nonostante i risultati meno ottimali sul training set. Probabilmente questo è dovuto al fatto che quello che considera anche la dummy *holidays* si adatta troppo bene ai dati in fase di learning.
Di conseguenza è stato il primo modello ad essere scelto per effettuare le previsioni finali.

4 UCM

Per quanto riguarda i modelli UCM, è stato deciso di considerare 3 diverse implementazioni della parte relativa al trend: un Local Linear Trend, un Integrated Local Linear Trend, e un Local Linear Trend stimato come Random Walk. Per quanto riguarda la parte stagionale, questa è stata modellata in modo uguale per le diverse istanzianzioni, attraverso delle dummies per la parte settimanale e attraverso 24 armoniche per quella annuale.

	MAPE sul Training set
UCM con LLT	0.07903713
UCM con ILLT	0.07905364
UCM con RW	0.07940868

Di seguito presentiamo i grafici e la tabella dei risultati dei diversi algoritmi sul validation set:

Figure 8: Confronto tra predizione UCM con LLT e dati reali sul validation set

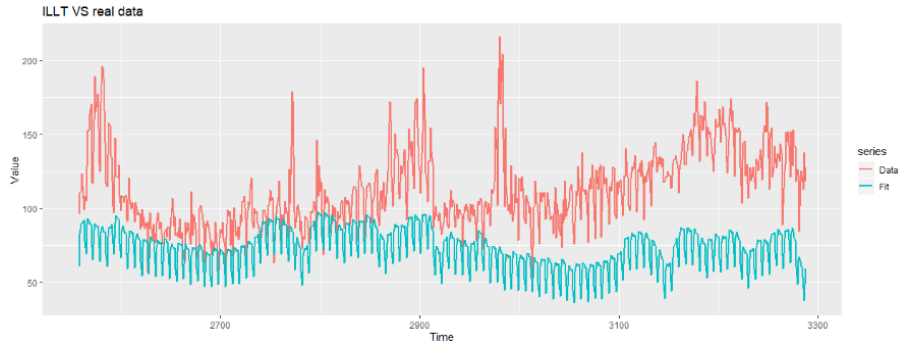

Figure 9: Confronto tra predizione UCM con ILLT e dati reali sul validation set

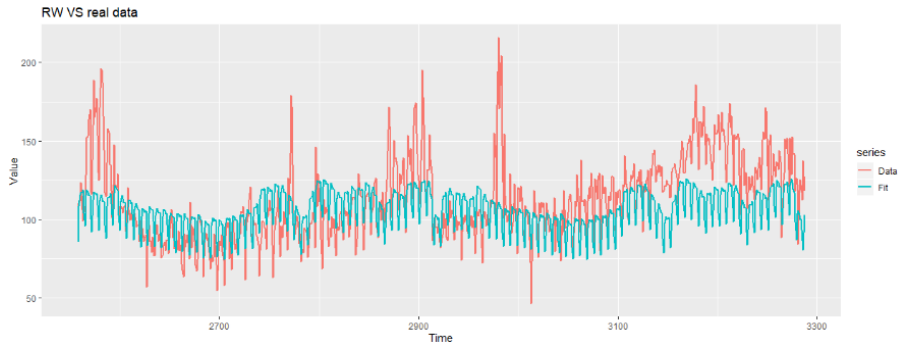

Figure 10: Confronto tra predizione UCM con RW e dati reali sul validation set

MAPE ottenuto dai diversi modelli sul validation set:

	MAPE
UCM con LLT	0.1543105
UCM con ILLT	0.3314398
UCM con RW	0.1500685

Per le previsioni finali è stato scelto il modello UCM con la componente trend stimata come Random Walk, date le performances migliori ottenute sul validation set.

5 Machine Learning

Per quanto riguarda le tecniche non lineari, si è deciso di procedere considerando due diversi tipi di modelli, il K-Nearest Neighbours Algorithm e le Recurrent Neural Networks. Per quanto riguarda la prima tipologia, sono state provate due diverse strategie per la previsione *multi-step ahead*: l'approccio **MIMO** (Multiple Input Multiple Output), in cui si predice in anticipo un periodo di tempo corrispondente al numero di lags, e l'approccio **recursive**, in cui viene sfruttata la tecnica di one-step ahead per compiere previsioni. Nel primo caso il numero di lags scelto è stato di 730, mentre nel secondo di 365. Inoltre, per entrambe le istanziazioni è stato cosniderato un numero di Nearest Neighbours tipicamente pari a $\sqrt{N} = 50$, dove N è il numero di osservazioni del training set.

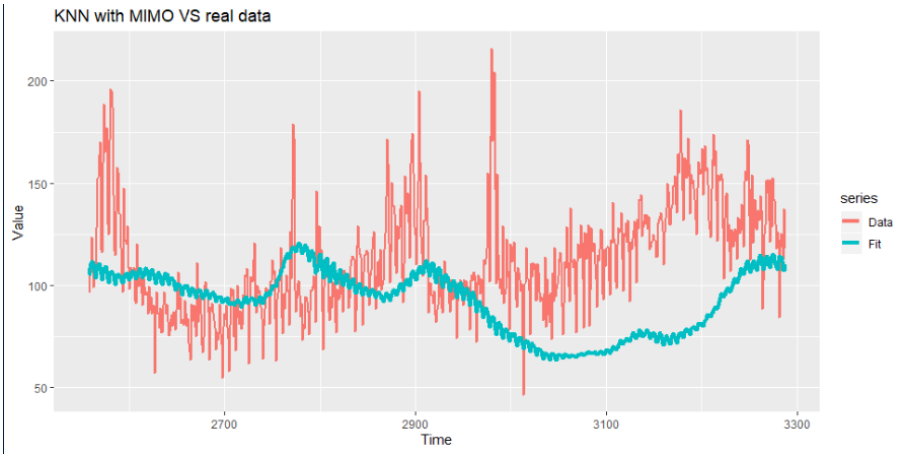

Figure 11: Confronto tra predizione KNN con *MIMO* e dati reali sul validation set

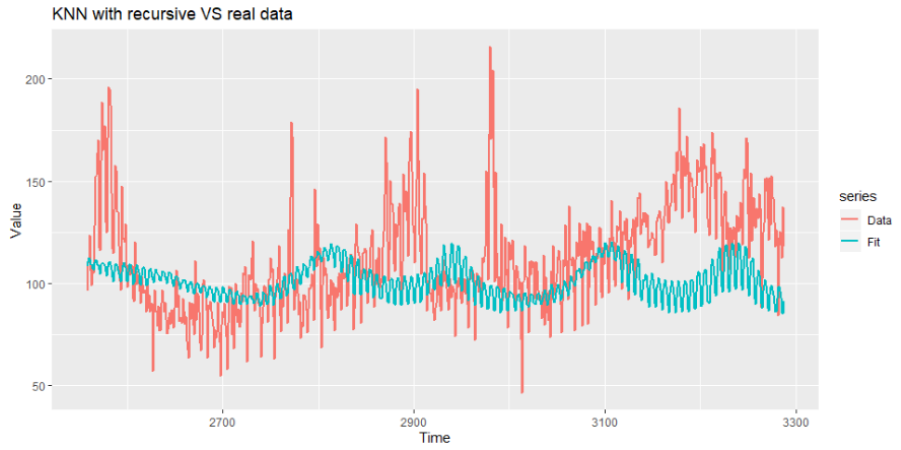

Figure 12: Confronto tra predizione KNN con *recursive* e dati reali sul validation set

Mape sul validation:

	MAPE
KNN con MIMO	0.2398395
KNN con recursive	0.1762849

Per quanto riguarda invece le Recurrent Neural Network, è stato deciso di sfruttare due diversi tipi di rete grazie alla libreria *Keras*: la Long Short Term Memory (LSTM) e la Gated Recurrent Unit (GRU). Per procedere all'implementazione di tali modelli è stata necessaria una fase di pre-processing dei dati, in cui questi sono stati centrati e scalati, per rendere più veloce la fase di learning.
La prima rete consta di un layer LSTM di 100 neuroni, un secondo layer LSTM di 40 neuroni, un terzo layer LSTM sempre di 40 neuroni e infine un layer denso. Similmente la seconda rete comprende un layer gru di 100 neuroni, altri due gru di 40 neuroni e uno denso. Per entrambi gli algoritmi, il numero di iterazioni per la fase di training è stato fissato a 50, la funzione di perdita considerata è il *Mean Absolute Error*, mentre *adam* è stato scelto come ottimizzatore.

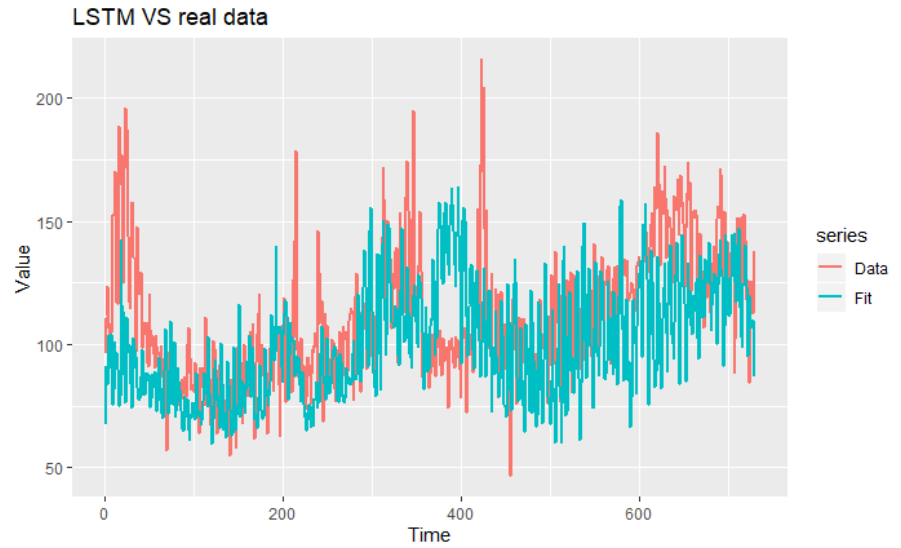

Figure 13: Confronto tra predizione della rete LSTM e dati reali sul validation set

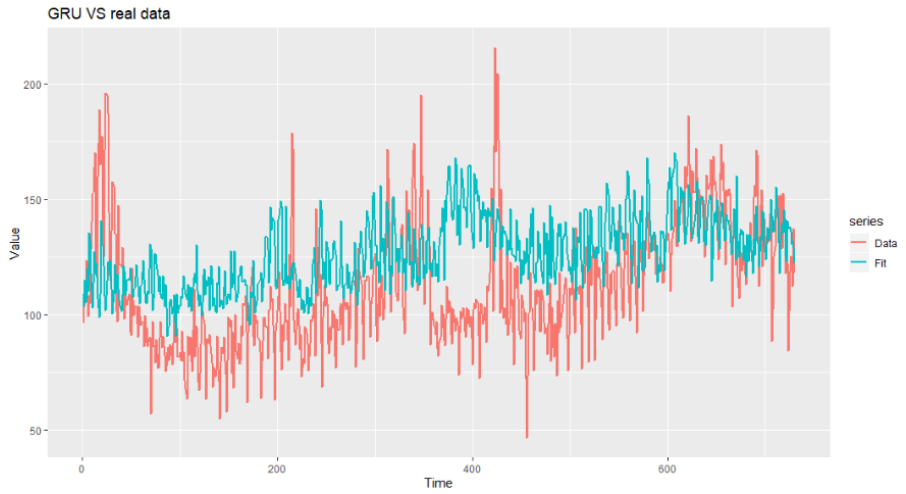

Figure 14: Confronto tra predizione della rete GRU e dati reali sul validation set

Mape sul validation:

	MAPE
LSTM	0.1923482
GRU	0.2339122

Dal raffronto dei risultati ottenuti dai vari modelli impiegati, è stato deciso di impiegare il K-Nearest Neighbours inizializzato con il parametro *recursive*, per effettuare le previsioni finali.

6 Previsioni dei dati dal 1 Gennaio al 31 Dicembre 2019

Di seguito si presentano i grafici delle previsioni ottenute con i migliori modelli per ciascuna categoria.

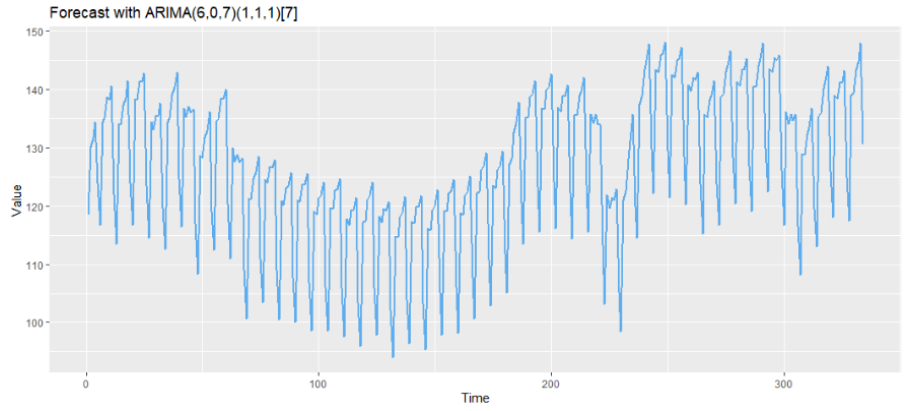

Figure 15: Previsione di ARIMA(6,0,7)(1,1,1)[7] con armoniche

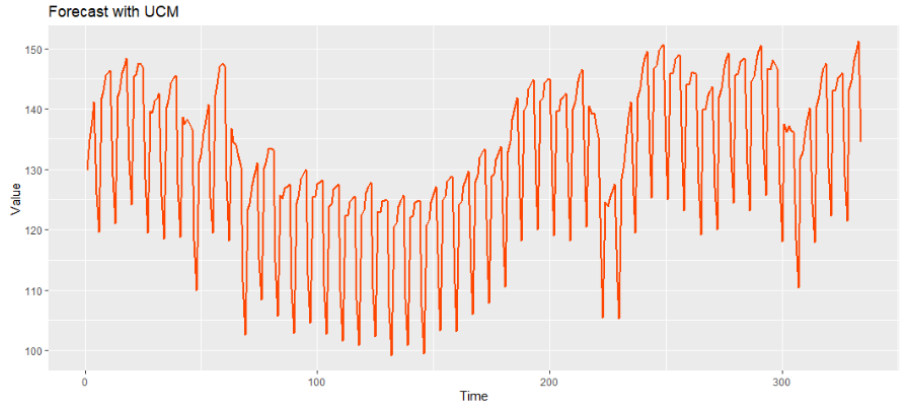

Figure 16: Previsione di UCM con Random Walk

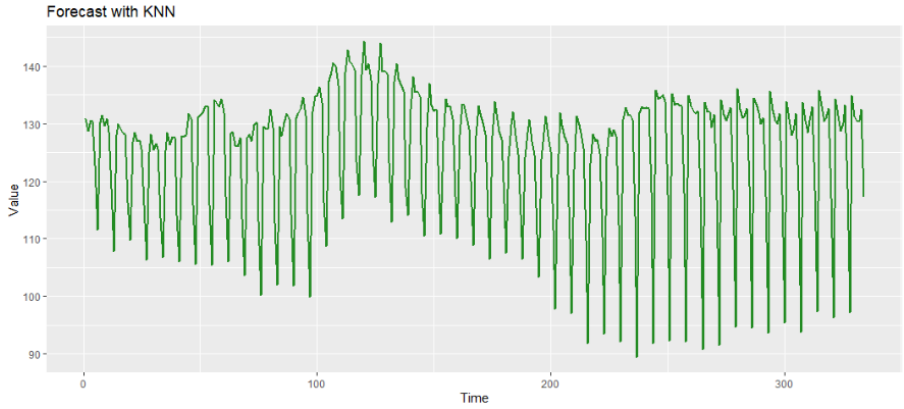

Figure 17: Previsione di KNN con *recursive*

7 Considerazioni Finali

I risultati ottenuti hanno mostrato che i modelli che sono riusciti a prevedere meglio i dati del validation set sono stati quelli relativamente più semplici e flessibili, soprattutto per quanto riguarda gli ARIMA e i modelli non lineari. Questo risulta evidente dal confronto sul test set delle performances dell'ARIMA con le sole armoniche e dell'ARIMA comprendente anche le dummies per le vacanze. Una considerazione ulteriore va fatta riguardo al valore di MAPE che i tre algoritmi migliori hanno ottenuto; infatti, mentre i migliori ARIMA e UCM hanno raggiunto rispettivamente MAPE 0.1566 e 0.1500, il miglior modello di Machine Learning ha raggiunto MAPE 0.1763, distanziandosi in modo abbastanza evidente dagli altri due anche confrontando graficamente le previsioni finali. Ci si aspetterà che probabilmente quest'ultimo ottenga risultati peggiori degli altri due sui dati predetti del 2019.
Infine, ulteriori miglioramenti per gli algoritmi provati sarebbero possibili, ad esempio provando un numero maggiore di inizializzazioni o architetture dei modelli; oltre a ciò potrebbe essere utile considerare l'aggiunta di ulteriori informazioni e metadati quali variabili o regressori che possano aiutare nella previsione dei dati futuri.