Università degli Studi di Milano Bicocca
Dipartimento di Informatica, Sistemistica e Comunicazione
Corso di laurea magistrale in Data Science

STREAMING DATA MANAGEMENT
AND TIME SERIES ANALYSIS
Progetto finale

Previsione Serie Storica di Monossido di Carbonio

Autore:
Leonardo Riva - 830647 - l.riva37@campus.unimib.it

Gennaio 2022

1 Introduzione

L'obiettivo del progetto è sviluppare un modello in grado di predire dei dati di una serie storica. Verranno sfruttate tre tipologie di algoritmi: i modelli lineari ARIMA e UCM e i modelli non lineari di machine learning kNN, LSTM e GRU. Ne verranno analizzate e confrontate le performance, al fine di determinare il modello migliore per lo scopo dell'analisi, riportando le decisioni prese nel definirlo.

2 Dataset

Il dataset utilizzato è costituito dalla serie storica oraria di misurazioni di CO, in un periodo dal marzo 2004 al febbraio 2005, per un totale di 8526 osservazioni. L'obiettivo è prevedere i 744 valori orari del mese di marzo 2005.

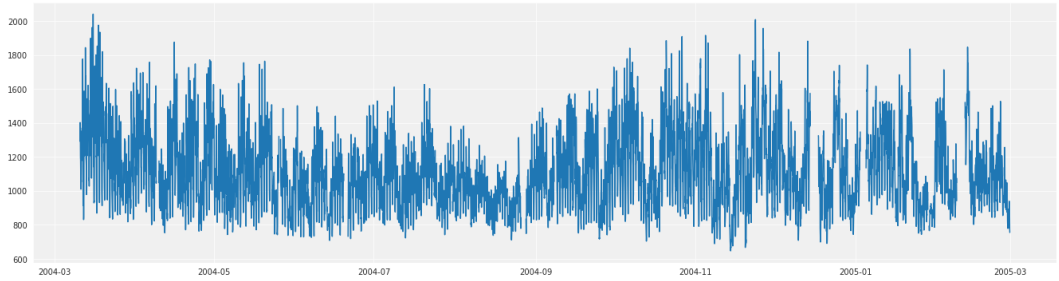

Figura 2.1: Serie storica iniziale

Non si nota nessun outlier evidente dovuto ad errori di misurazione, ma, al contempo, la serie presenta 365 missing values. Per far fronte a questo problema, ognuno di esso è stato sostituito dalla media dei 3 valori precedenti alla stessa ora e stesso giorno della settimana.

Successivamente, il dataset è stato diviso in training set e validation set. Si è scelto di considerare come validation set l'ultimo mese di dati a disposizione in modo da mantenere un buon numero di osservazioni per il training. Questo porta ad un training set di 7854 osservazioni (92.1%) ed un validation set di 672 osservazioni (7.9%).

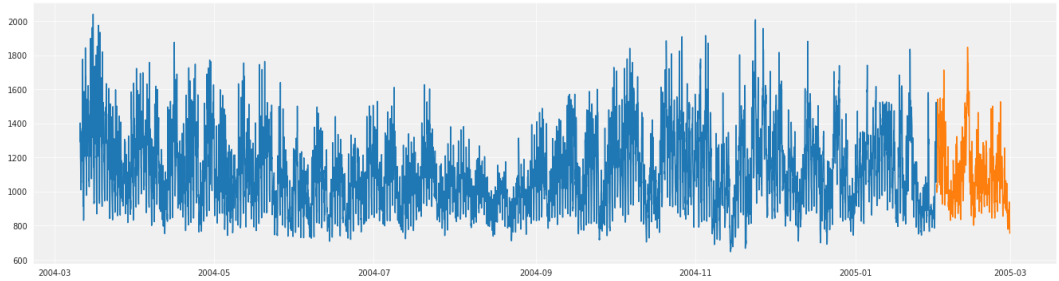

Figura 2.2: Suddivisione in training set e validation set della serie senza missing values

Esplorando più a fondo i dati, si può notare la presenza di più stagionalità: da una parte giornaliera, con una rilevazione inferiore di CO nelle ore notturne, mentre dall'altra settimanale, con una rilevazione inferiore nei week end.

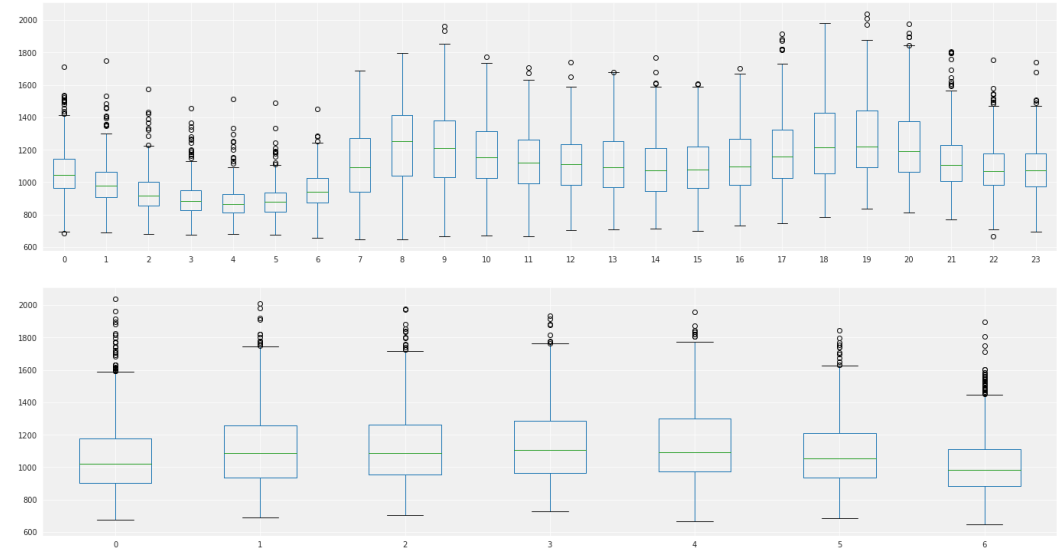

Figura 2.3: Boxplot che descrivono stagionalità rispettivamente giornaliera e settimanale.

3 Procedimento

Come già descritto, verranno implementati diversi approcci. Per ciascuno di questi, per valutarne le performance predittive, è stata utilizzata la metrica del Mean Absolute Percentage Error (MAPE), dato dalla media aritmentica dei rapporti tra il valore assoluto degli errori di previsione e il valore reale; è inoltre facilmente interpretabile dal punto di vista umano e permette di confrontare i diversi modelli.

3.1 ARIMA

Per prima cosa è stata analizzata la stazionarietà. È stata osservata una correlazione lineare tra varianza e media a diversi mesi, con R=0.86 e p-value=0.0003, suggerendo l'utilizzo del logaritmo sulla serie. Il test di Augmented Dickey-Fuller, con un p-value di 0.20, conferma inoltre che la serie non è stazionaria in media, e necessita di una differenziazione.

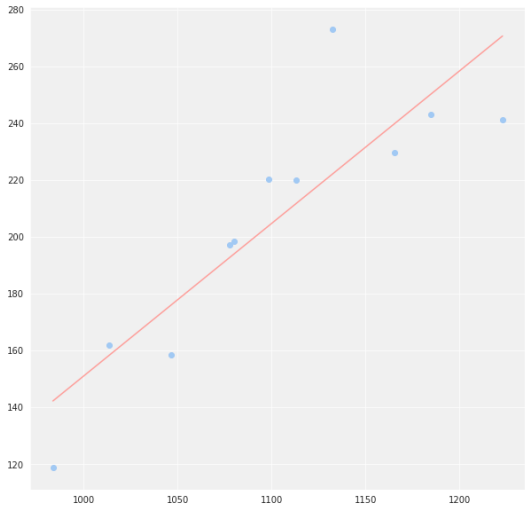

Figura 3.1: Correlazione lineare tra varianze e medie.

Osservando invece i grafici ACF e PACF della serie iniziale, si nota una chiara stagionalità ogni 24 osservazioni (1 giorno), come già notato nella fase iniziale di esplorazione del dataset.

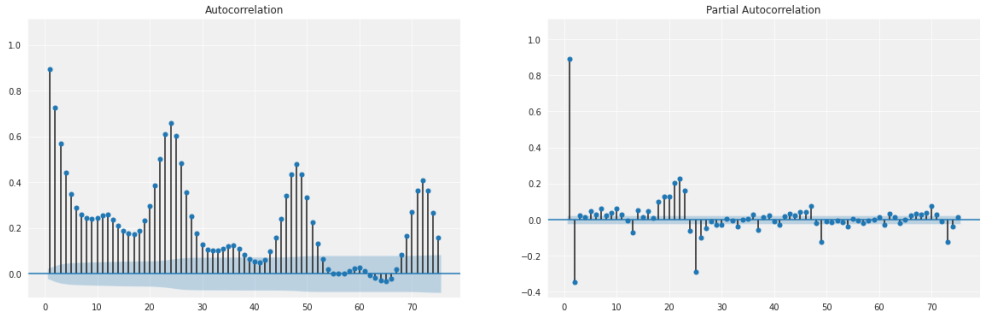

Figura 3.2: ACF e PACF iniziali.

Di conseguenza, è stato creato un primo modello, sul logaritmo della serie storica: $ARIMA(0,1,0)(0,1,0)[24]$. Analizzando i suoi residui, si nota come sia necessaria l'aggiunta di un $SMA(1)$, come di solito accade differenziando.

Viene costruito quindi il modello: $ARIMA(0,1,0)(0,1,1)[24]$. I suoi residui sono migliorati, avendo valori più contenuti nelle bande di confidenza, ma rimane della non stocasticità nei residui stagionali giornalieri. Purtroppo, non è stato individuato

un modo per ridurli ulteriormente. Il MAPE sul validation set ha un valore di 11.87%, mentre un AIC di -19660.

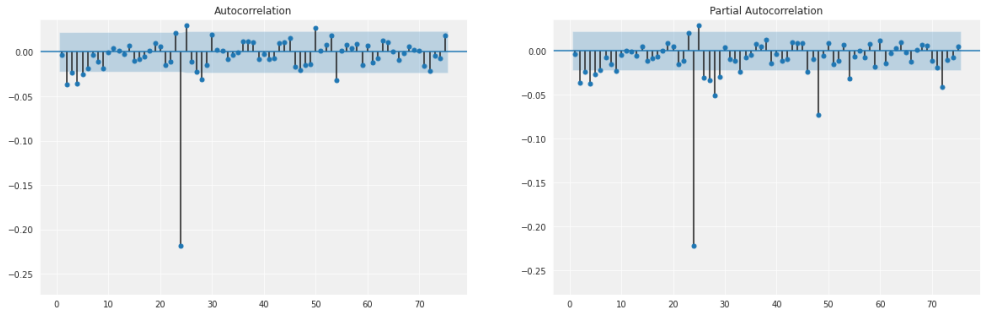

Figura 3.3: ACF e PACF dei residui del modello $ARIMA(0,1,0)(0,1,1)[24]$.

Per quanto riguarda i residui a lag inferori, non stagionali, essi non sono direttamente interpretabili a dei valori di p e q. Per questo, è stata fatta una grid-search per trovare il modello che modellasse meglio i dati, in termini di AIC e MAPE. I valori risultanti sono $p = 2$ e $q = 2$, con un MAPE di 11.52% e un AIC di -20212.

Si giunge quindi al modello: $ARIMA(2,1,2)(0,1,1)[24]$. Rimane ancora da modellare la stagionalità settimanale, chiaramente visibile nei grafici ACF e PACF a lag più alti. Questa viene modellata utilizzando delle sinusoidi, a periodo 168.

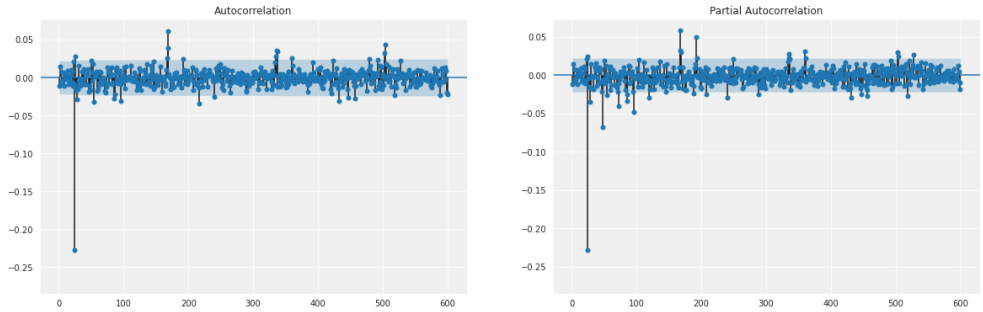

Figura 3.4: ACF e PACF dei residui del modello $ARIMA(2,1,2)(0,1,1)[24]$, questa volta a 600 lag.

Dopo un test per verificare quale fosse il numero ottimale di sinusoidi, ovvero 3, è stato creato il modello. Di nuovo, c'è un miglioramento dell'AIC a -20249 e del MAPE sul validation set a 10.94%.

Un'ultima versione consiste nell'aggiungere dei regressori dummy per modellare le festività, le quali possono avere dei valori anomali. Dopo un'analisi (manuale) sulla serie storica, si è notato come in poche di essere si verificano eventi fuori dal comune, come il 25 aprile, 2 giugno, 25 dicembre, ecc. Il loro inserimento, in realtà, non comporta miglioramenti in termini di MAPE e di AIC.

In conclusione, il miglior modello ARIMA presenta un AIC di -20249 e un MAPE di 10.94%.

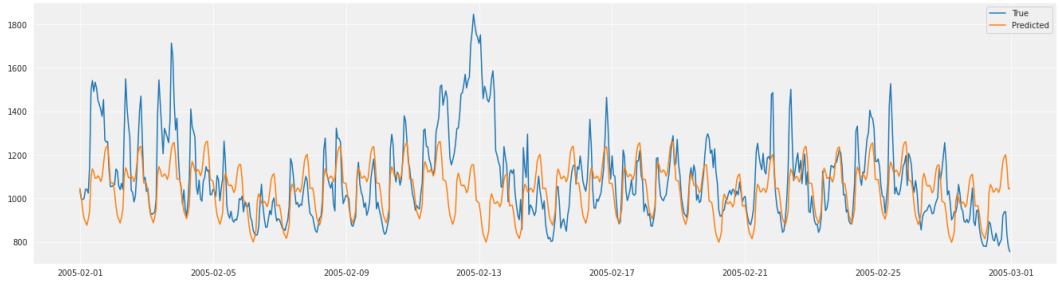

Figura 3.5: Previsione finale del modello ARIMA.

Dalla figura 3.5 si può notare come la stagionalità settimanale sia perfettamente catturata, un po' meno la giornaliera, che è meno stabile nel tempo. Inoltre, i valori anormali del 12-13 febbraio non vengono catturati.

Infine, individuato il modello più performante, viene verificato che i suoi residui siano generati da un processo white noise, quindi a media nulla, varianza costante e incorrelati con il proprio passato. Mentre i primi due criteri vengono rispettati, l'ultimo non completamente; rimane, infatti, della memoria significativa non modellata. Questo può anche essere causato dal fatto che, trattandosi di dati reali, è difficile ottenere dei residui white noise perfetti.

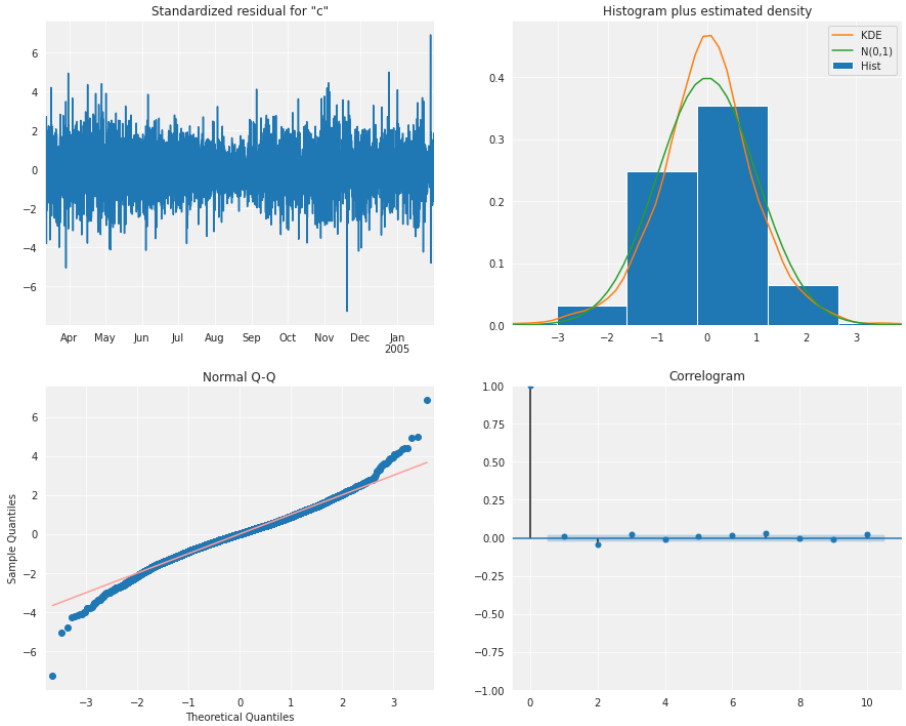

Figura 3.6: Riepilogo sui residui del modello.

3.2 UCM

Per questa fase si è deciso di iniziare l'analisi con un modello semplice: conoscendo già i tipi di stagionalità, ne sono state aggiunte una giornaliera dummy e una settimanale sinusoidale, mentre per il level sono stati diversi tipi (random walk, local linear trend, ecc), per determinare il più adatto.

Individuato il random walk with drift, si è poi passati a verificare se modellare le stagionalità in quel modo fosse effettivamente ottimale. Infatti, si è capito come il modo migliore fosse vederle entrambe scritte come sinusoidi. In particolare, dopo una serie di test, il numero ottimale di sinusoidi è di 5 e 3, rispettivamente per stagionalità stocastica giornaliera e settimanale. È importante notare che, al fine di evitare che l'algoritmo non raggiunga la convergenza, è stata impostata la varianza iniziale del modello ad un valore più consono, cioè la varianza della serie storica di training.

L'aggiunta della componente di ciclo stocastico non ha portato miglioramenti al modello. Lo stesso vale per l'aggiunta dei regressori esterni dummy sulle festività, così come accadeva anche in ARIMA.

Quindi, il modello che ottiene le migliori performance, in temini di AIC e MAPE (compromesso tra i due), è quello composto da:

- random walk con drift
- 6 sinusoidi stocastiche per modellare la stagionalità giornaliera
- 2 sinusoidi stocastiche per modellare la stagionalità settimanale

Questo modello ottiene un AIC di -19414 e un MAPE di 11.13%. Si notano, anche graficamente, le minori performance rispetto alla previsione ARIMA.

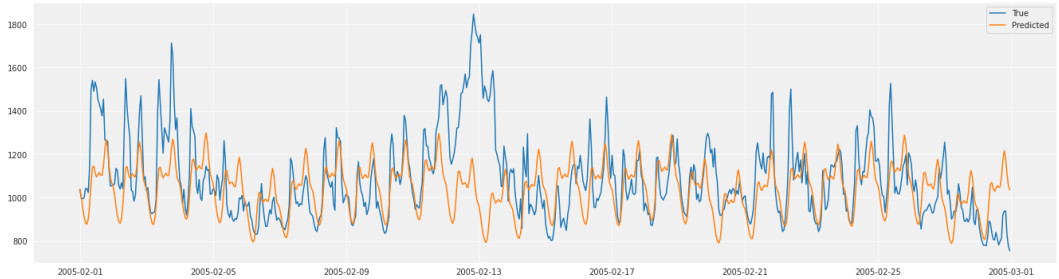

Figura 3.7: Previsione finale del modello UCM.

I residui del modello finale sono simili a quelli di ARIMA, mantenendo purtroppo ancora più memoria e correlazione.

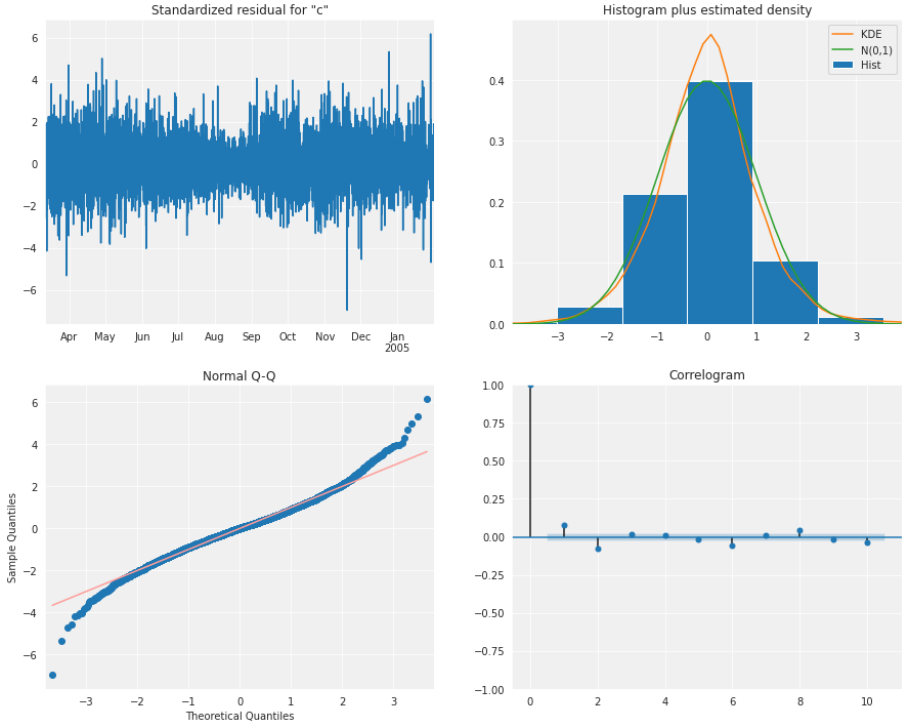

Figura 3.8: Riepilogo sui residui del modello.

3.3 Machine Learning

Sono state implementate diverse tecniche: K-Nearest-Neighbors e Recurrent Neural Networks (Long Short-Term Memory e Gated Recurrent Network)

3.3.1 KNN

L'algoritmo prevede i valori futuri basandosi sui k Nearest Neighbors. Viene utilizzata una metodologia MIMO (multi-input multi-output).

Per la scelta dei valori degli iperparametri k (numero di osservazioni passate da considerare) e p (quante sottosequenze simili guardare) è stata effettuata una serie di test, andando a prendere quelli che minimizzassero il MAPE sul validation set. I valori testati sono: k in [4, 8, 16, 32, 64] e p in [12, 24, 24*2, 24*4, 24*7, 24*14, 24*21, 24*28]. Si sarebbe potuta effettuare una grid-search più intensa, ma è stato preferito in questo modo al fine di avere una maggiore interpretabilità e velocità di esecuzione. Inoltre, usare la mediana ha fornito risultati leggermente migliori della media.

I migliori risultano essere $k = 8$ e $p = 504$ (tre settimane), ottenendo un MAPE di 10.13%.

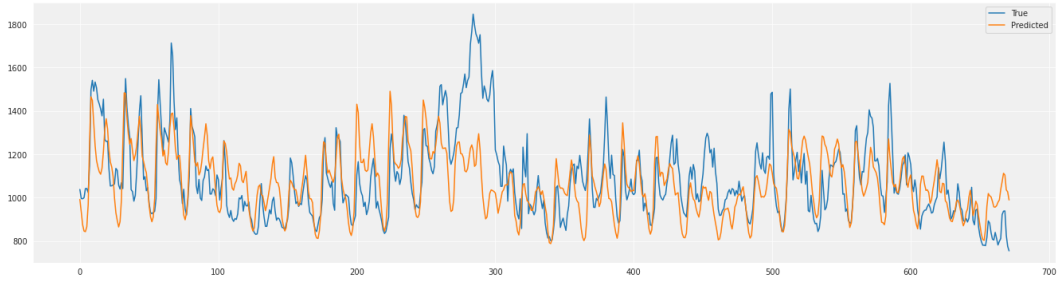

Figura 3.9: Previsione finale del KNN.

3.3.2 LSTM

Per prima cosa, i dati sono stati sottoposti a del preprocessing: oltre ad una normalizzazione tra 0 e 1 (ovviamente rispetto al training set), hanno subito un reshape per essere dati in input alla rete. In particolare, è stata scelta una previsione recursive multi-step, predicendo una settimana alla volta. Questa tecnica fornisce risultati migliori rispetto ad una single-step o ad una multi-step giornaliera. La finestra ottimale sui dati precedenti è di tre settimane.

Sono state effettuate inoltre diverse prove sulla struttura del modello. Il migliore, che cerca di non essere troppo complesso per un trade-off sul tempo di allenamento (i modelli lineari sono estremamente più veloci), è così costruito:

- LSTM di 512 unità, con funzione di attivazione ReLU
- Dropout di 0.1
- LSTM di 256 unità, con funzione di attivazione ReLU
- Dropout di 0.1
- Dense di 168 unità (per predizione valori della settimana successiva)

Per quanto riguarda il numero di epoche, è stato effettuato un early stopping con patience=3, su 100 epoche, con un batch size di 128 osservazioni.

La previsione avviene settimana per settimana: prendendo in input l'ultimo mese del training set, predice una settimana; aggiorna poi i dati prendendo tre settimane del training set e quella predetta; così via fino ad arrivare ad un mese. Il modello con il MAPE minore risulta essere 10.43%.

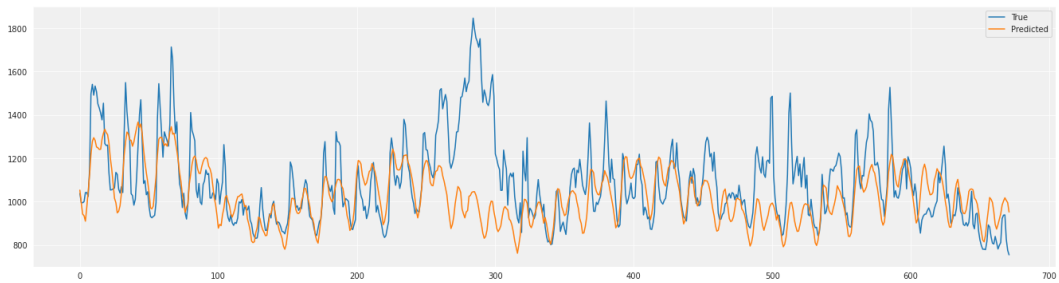

Figura 3.10: Previsione finale della RNN con LSTM.

3.3.3 GRU

Per le GRU, i dati subiscono lo stesso preprocessing delle LSTM. Anche la struttura della rete viene mantenuta, sostituendo però i layer LSTM con dei layer GRU. Il procedimento è lo stesso in tutto e per tutto, in quanto sembra essere la migliore impostazione anche per questo tipo di RNN. Il modello ottiene un MAPE di 10.93%

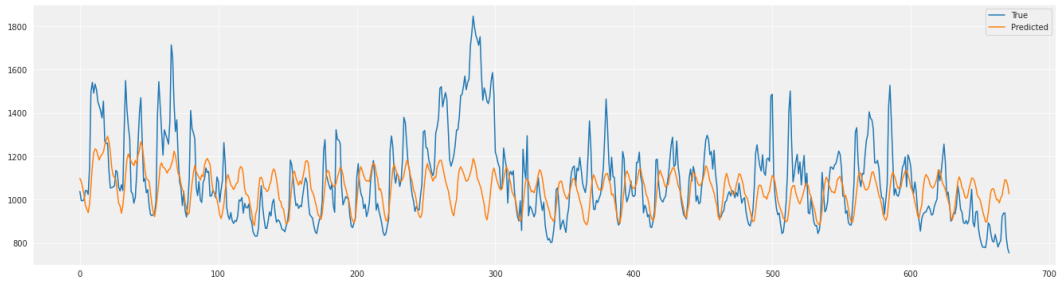

Figura 3.11: Previsione finale della RNN con GRU.

4 Conclusioni

Tenendo in considerazione la metrica MAPE sul validation set, il modello che ha mostrato i migliori risultati, è il KNN (che rimane pur sempre model-free), seguito dai modelli lineari. Questi ultimi non performano al meglio, ma questo può essere dovuto alla non corretta impostazione dei parametri (i resuidi dei modelli finali avrebbero potuto essere migliorati). L'aspettativa è quindi che il KNN si confermi il più preciso anche nella previsione del mese successivo.

ARIMA	UCM	RNN
10.94%	11.13%	10.13%

Tabella 4.1: Risultati del modello migliore per ogni categoria.

In conclusione, i modelli vengono riaddestrati su tutta la serie storica a disposizione, in modo da effettuare una previsione sul mese di marzo 2005.

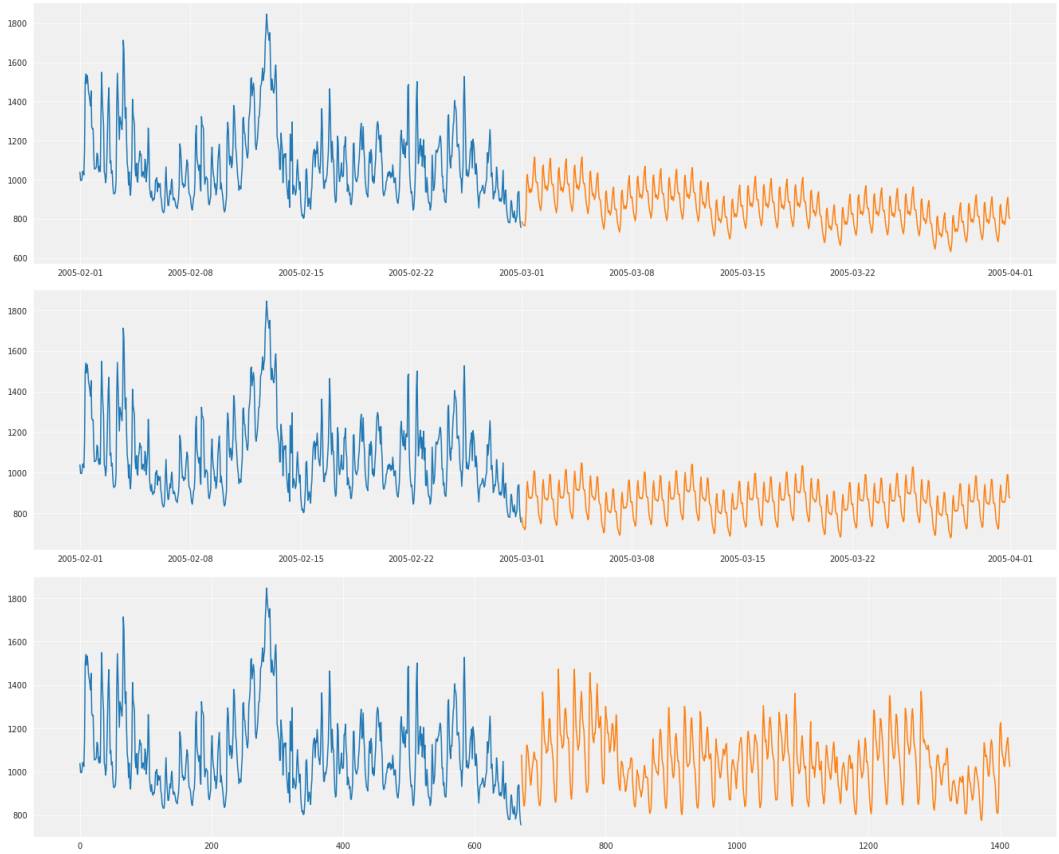

Figura 4.1: Previsioni su marzo 2005 dei modelli ARIMA, UCM e KNN.