PROGETTO
FORECAST TIME SERIES STREAMING DATA

a cura di Matteo Paparella, matr. 812561

Anno Accademico 2020/2021

INDICE

- Pag. 1 – Introduzione;
- Pag. 2 – Data Pre-Processing iniziale;
- Pag. 2 – Data Exploration;
- Pag. 5 – Data Preparation;
- Pag. 5 – Modello 1: SARIMAX;
- Pag. 12 – Modello 2: UCM;
- Pag. 16 – Modello 3: PROPHET;
- Pag. 19 – Modello 4: LSTM;
- Pag. 21 – Conclusioni;

INTRODUZIONE

Il dataset a disposizione è composto inizialmente dalle seguenti features:

- "DATA": prende in considerazione un arco temporale dal 01/09/2018 al 31/08/2020 di rilevazioni;
- "Ora": prende in considerazione l'ora di rilevazione;
- "VALORE": prende in considerazione il valore della rilevazione come dato numerico;

Non è fornita un'informazione sul contesto nel quale i dati sono stati rilevati e come.
Lo scopo del progetto è quello di effettuare una predizione per il periodo temporale dal 01/09/2020 0:00:00 al 31/10/2020 23:00:00 utilizzando modelli di predizione di dati basati su time series. Dovranno essere utilizzati sia modelli lineari che modelli non lineari (machine learning) tenendo il MAE come misura dell'errore, quindi andando a minimizzare il MAE per quanto riguarda le previsioni sul training e validation set.

Per MAE si intende il Mean Square error, calcolato come l'errore medio assoluto tra il valore predetto e quello effettivo, dove il valore assoluto è calcolato introdotto per evitare che i valori si cancellino a vicenda.

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}.$$

DATA PRE-PROCESSING INIZIALE

Il dataset iniziale necessita di alcune operazioni di pre-processing:

- Cambio del valore delle Ore in scala 0-23 da 1-24, che era il formato iniziole dato;
- Costruzione colonna "DataeOra" che comprende la data e l'ora in formato "%Y-%m-%d-%H";
- Utilizzo della colonna "DataeOra" come nuovo index per il dataset complessivo che sarà utilizzato per le analisi;
- Estrazione e creazione colonne con numero mese, giorno, anno;
- Mancano valori del 2019-03-31 03:00:00 e 2020-03-29 03:00:00 che vengono sostituiti con il valore dell'ora precedente;
- Mancano i valori del 2020-05-31, è dunque utilizzata la media dei valori dei 14 giorni prima e 14 giorni dopo;

Il dataset così pre-processato viene esportato per una maggiore praticità per operazioni successive. Le rilevazioni dei dati mancanti sono state effettuate prendendo in considerazione il numero di giorni per mese ed il numero di ore per mese. Viene effettuato un conteggio per eventuali valori mancanti come ulteriore verifica che le operazioni siano state effettuate correttamente. Al termine delle operazioni di pre-processing, possiamo passare alla prima visualizzazione complessiva dei dati a disposizione:

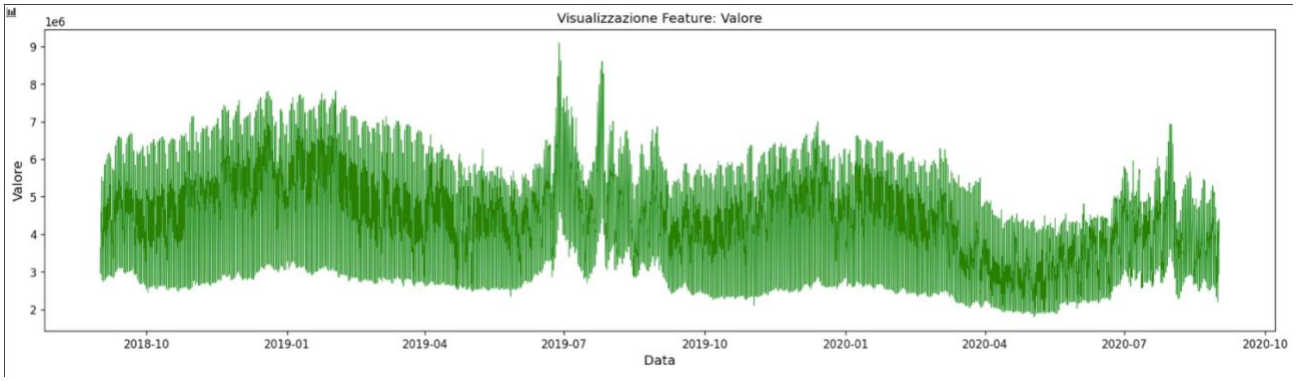


Figura 1: Visualizzazione complessiva dei valori a disposizione

Osservando la visualizzazione precedente è possibile osservare dei "picchi" nel periodo estivo ed un andamento annuale che presenta valori più alti nei mesi invernali avvicinandosi a gennaio ed una lenta riduzione nel procedere verso il periodo estivo. Possiamo quindi dedurre una stagionalità annuale, che voglio confermare. Inoltre, procedo con l'osservare l'eventuale presenza di altre stagionalità.

DATA EXPLORATION

Procedo ad osservare, con ulteriori visualizzazioni, la presenza o meno di ulteriori stagionalità nei dati. Per prima cosa cerco di confermare la mia ipotesi di stagionalità annuale. Con il confronto riportato in figura 2 posso constatare la presenza di una stagionalità annuale che caratterizza i dati a disposizione. La ricorrenza dello stesso andamento è chiara, da notare la presenza di una riduzione con il proseguire degli anni, in particolare si può osservare nei primi mesi del 2020, confrontandoli con i primi mesi del 2019.

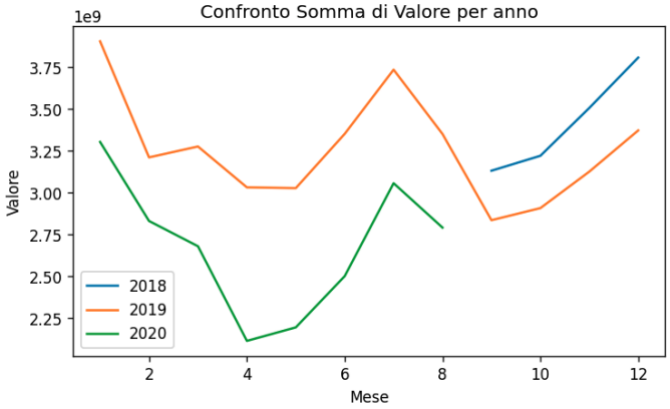
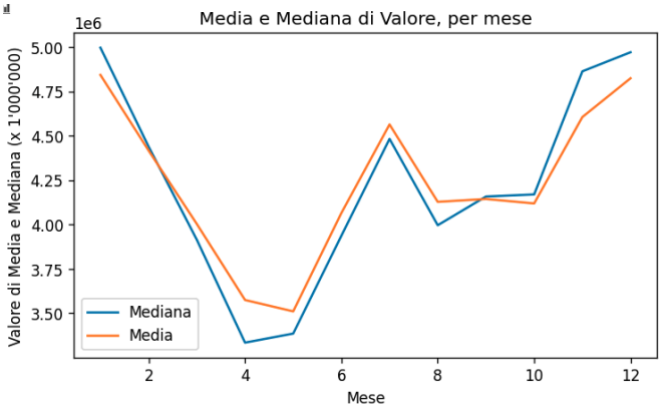


Figura 2: Visualizzazione andamento valore mensile confrontato per anno

Osservando la decrescita nei primi mesi del 2020 rispetto ai primi mesi del 2019, posso dedurre che la spiegazione sia il Covid-19, però osservo che la discrepanza di valori negli ultimi mesi del 2018 rispetto agli ultimi mesi del 2019 sia comunque rilevante, la differenza vera la si può osservare da metà marzo in poi, mesi di diffusione della preoccupazione da Covid-19.



Posso inoltre confermare la presenza di un andamento negativo verso i mesi estivi con un rialzo dei valori nel proseguire verso i mesi invernali. Si può osservare come il mese di luglio presenta valori mediamente più elevati rispetto ai mesi estivi, tenendo una media "invernale". Con la visualizzazione in figura 4 effettuo ulteriori considerazioni.

Figura 3: Visualizzazione media e mediana dei valori durante l'anno.

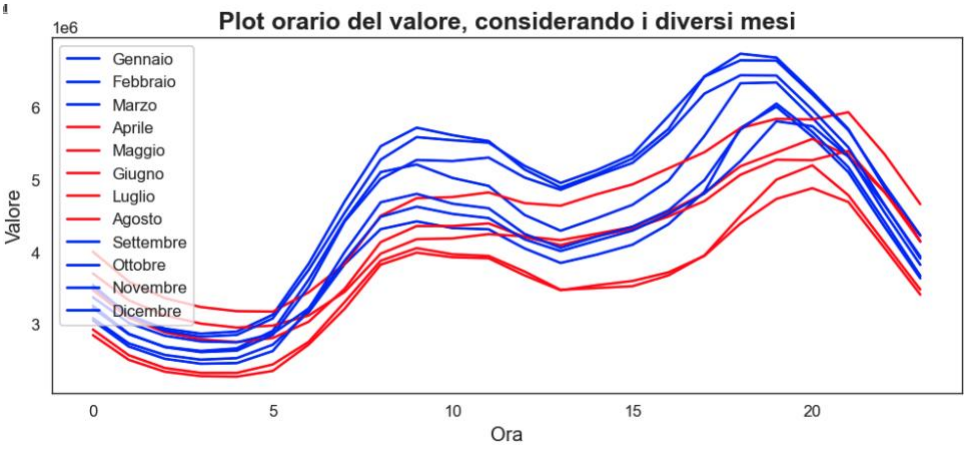


Figura 4: Visualizzazione valori andamento giornaliero suddiviso per mesi con legenda.

Osservo un abbassamento dell'andamento dei valori per i mesi di invernali (colore blu), rispetto ai mesi estivi (colore rosso), confermando le mie considerazioni precedenti. Inoltre, posso osservare la presenza di una stagionalità giornaliera con andamenti dei rialzi verso le ore 10 e le ore 19, per i mesi invernali, mentre per le ore 11 e le ore 21 per quanto riguarda i mesi estivi, probabilmente dovuto al cambio ore effettuato in primavera (portando l'orario avanti di 60 minuti).
Evidenziando il periodo dal 01/01 al 30/04 del 2019 e del 2020, osservo la netta riduzione dei valori, ho preso in considerazione l'ora 12 essendo abbastanza in linea con una media giornaliera di valore.

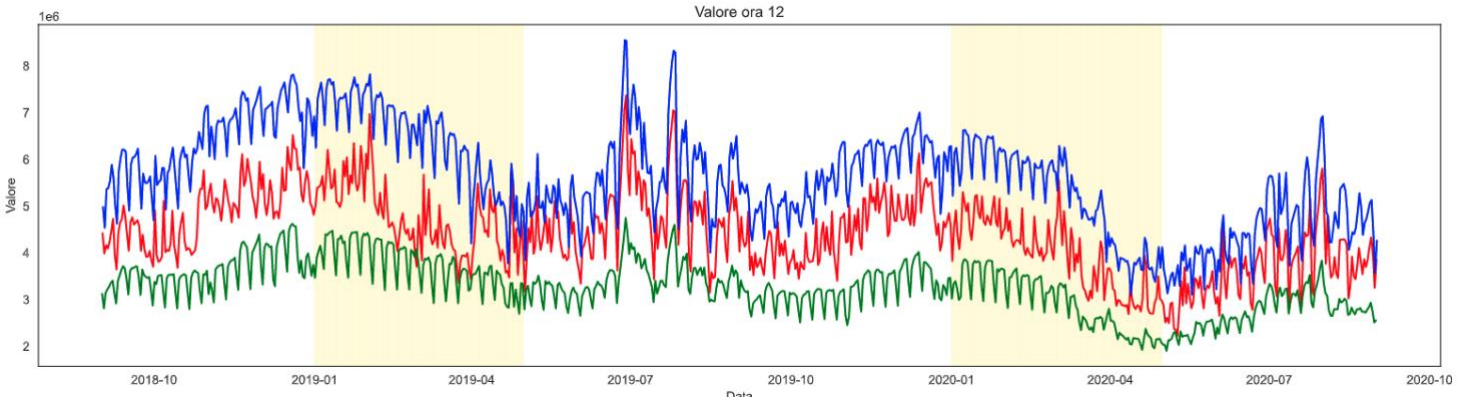


Figura 5: andamento valori ora 12 rispetto ai vari anni, evidenziando i primi 4 mesi dell'anno.

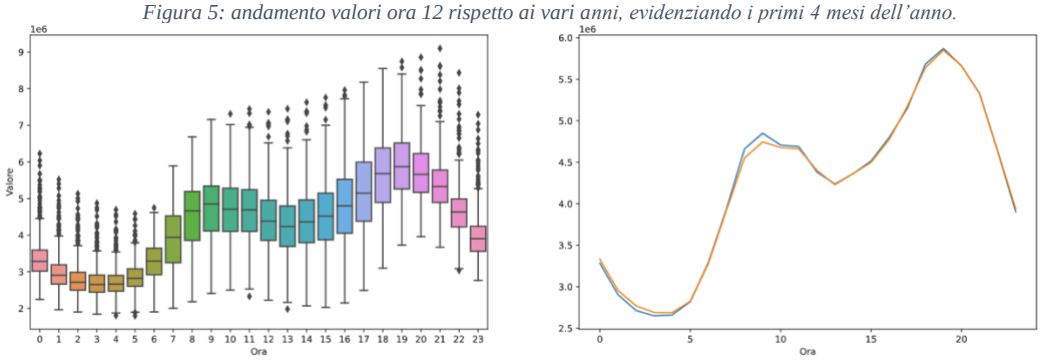
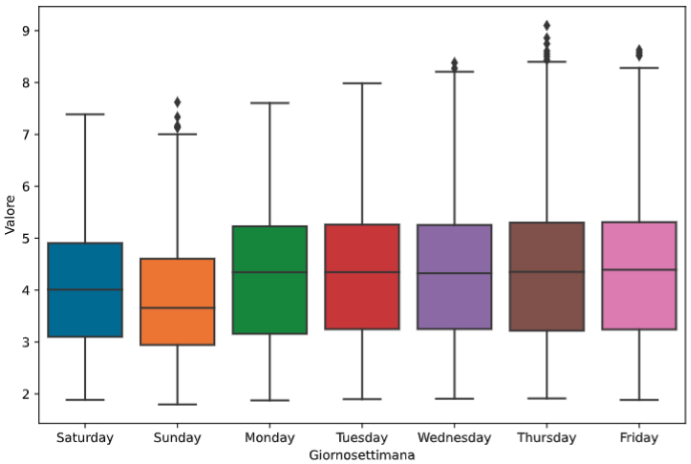


Figura 6: andamento orario dei valori, osservando i boxplot.
Figura 7: andamento orario calcolato come media e medina giornaliera, un'ulteriore visualizzazione per confermare i boxplot.

Osservo una stagionalità infrasettimanale, dovuta ai valori nel corso della settimana, con una netta diminuzione nel weekend. La conferma di ciò si può avere dai boxplot settimanali per giorno della settimana e dall'andamento della media e mediana per giorno della settimana, che presenta un "buco" nel giorno della domenica.



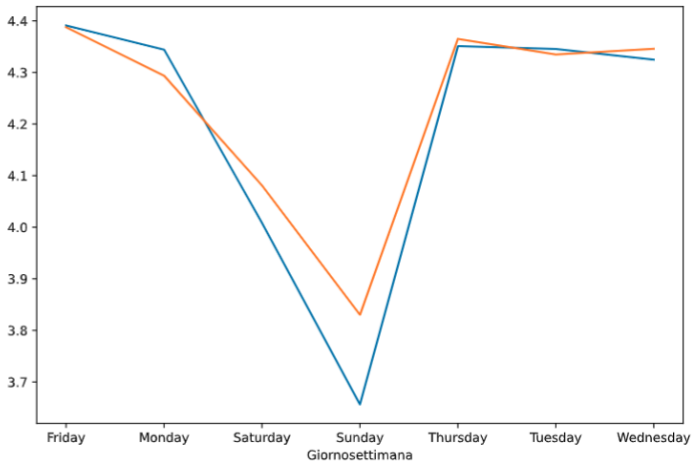


Figura 8: boxplot che raffigurano i valori per giorno della settimana.
Figura 9: andamento media e mediana dei valori dei giorni della settimana.

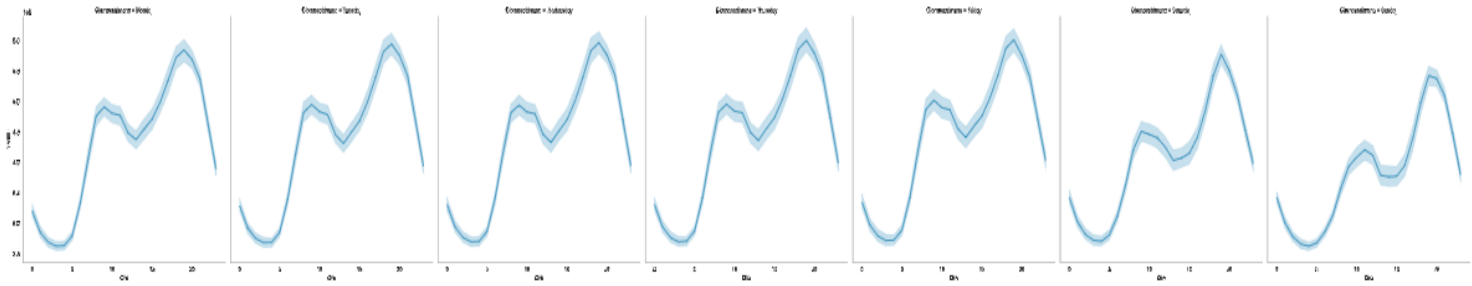


Figura 10: visualizzazioni raffiguranti l'andamento per ogni giorno, si osserva la riduzione nel weekend

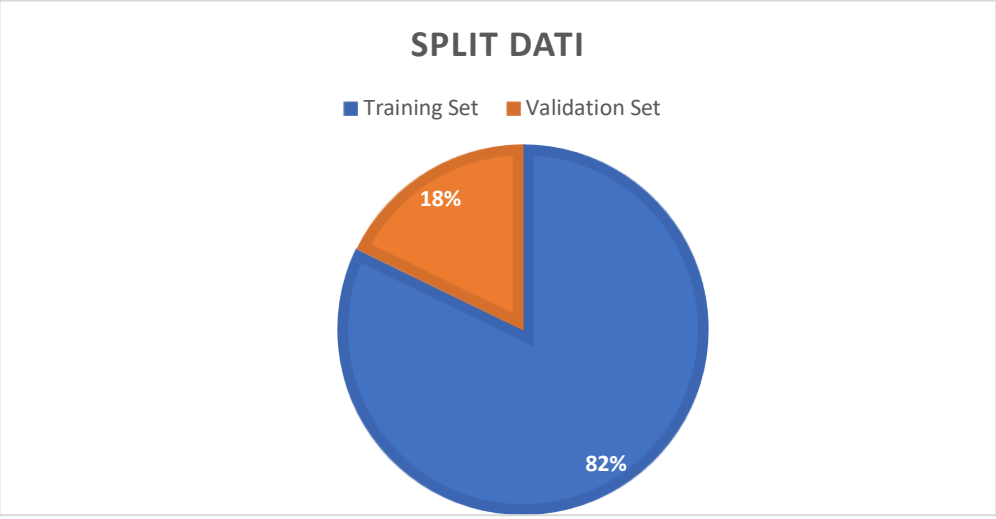
Ricapitolando, dalla fase esplorativa è emerso:

- Stagionalità annuale, con andamento in diminuzione dei valori durante il periodo estivo ed aumento nel periodo invernale, fatta eccezione per luglio;
- Stagionalità giornaliera, dovuta alle ore del giorno che presentano un andamento con "picchi" verso le ore 10 e 20 per i mesi invernali e 11 e 21 per i mesi estivi;
- Stagionalità infrasettimanale, dovuta alla presenza di una diminuzione dei valori per i giorni del weekend della settimana.

DATA PREPARATION

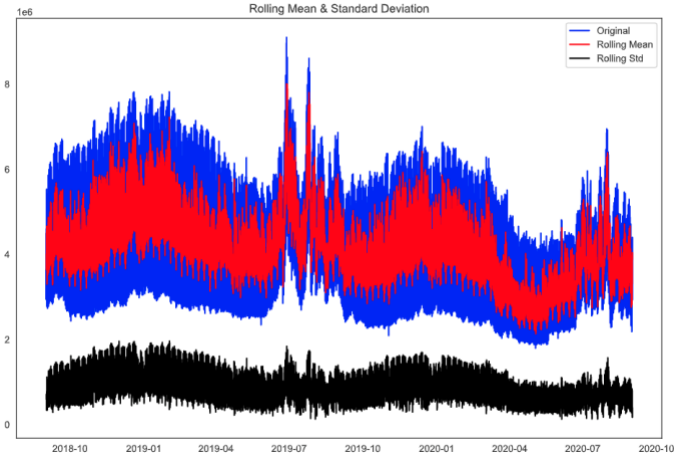
Nella fase che precede l'implementazione dei modelli, espongo le decisioni riguardo lo split del dataset in training set e validation set.
Nel prendere questa decisione, mi sono basato sull'andamento annuale degli ultimi mesi, osservando, come precedentemente esposto, che l'andamento dei valori era in diminuzione e che quindi ci fosse un "trend" negativo. Di conseguenza ho deciso di non utilizzare un classico split al 70% training e 30% validation, ma di basarmi sul tenere il maggior numero di valori nel training per dargli più dati possibili per allenarsi e calcolare questa diminuzione, e ovviamente attendendomi nelle previsioni che il modello colga la suddetta diminuzione. Nel corso della relazione utilizzerò il termine "validation set" per riferirmi al "test set", poiché è stato utilizzato per validare le mie considerazioni sui modelli utilizzati.
Di seguito, espongo la mia suddivisione:

Tipologia split	Range date	Numero rilevazioni	Percentuale corrispondente
Training Set	Dal 01/09/2018 00:00:00 al 23/04/2020 23:00:00	14424	82,21%
Validation (Test) Set	Dal 24/04/2020 00:00:00 al 31/08/2020 23:00:00	3120	17,79%
Complessivo		17544	100%



MODELLO 1: SARIMAX

Come primo modello, utilizzo i modelli ARIMA, valutando come misura dell'errore delle previsioni sul training e validation il MAE, con la finalità di minimizzarlo. Prima di applicare qualsiasi modello statistico di serie temporale, e particolarmente per i modelli ARIMA che lo richiedono, voglio assicurarmi che sia stazionaria, o per lo meno osservarne la presenza o meno di stazionarietà. Ricordo che i dati sono ritenuti stazionari se la media della serie temporale non aumenta nel tempo e, generalmente, non è possibile osservare dei pattern nel lungo periodo.
Effettuo delle prime considerazioni:



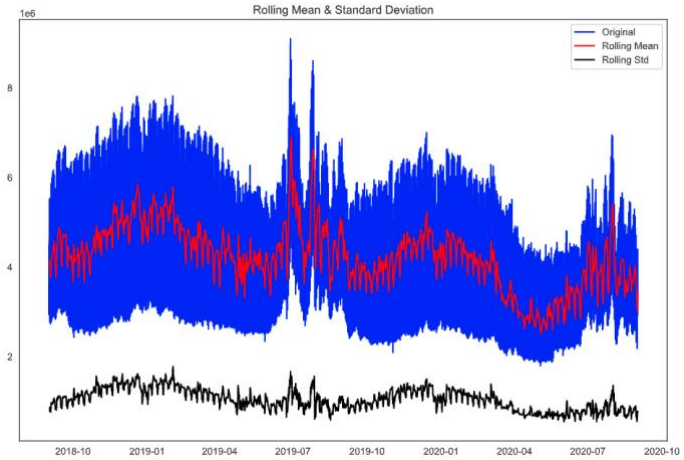


Figura 11: media e standard deviation prendendo in considerazione 12 rilavazioni alla volta.
Figura 12: media e standard deviation prendendo in considerazione 24 rilavazioni alla volta.

Osservo come la media giornaliera sia abbastanza costante nel tempo eccetto per i mesi estivi dove sono presenti i "picchi" intorno al mese di luglio. Continuo ad osservare la stazionarietà.
Divido in due *range* i dati e calcolo la media e varianza delle due partizioni, osservando come dividendo in due finestre i dati, la media e la varianza non rimangono propriamente costanti.
Ecco di seguito i risultati:

- Media range 1: 4627699.066689;

- Media range 2: 3839093.460214;
- Varianza range 1: 1607061437346.171387;
- Varianza range 2: 1251632563571.640869;

Effettuo il calcolo della "rolling mean" applicando ai miei dati:

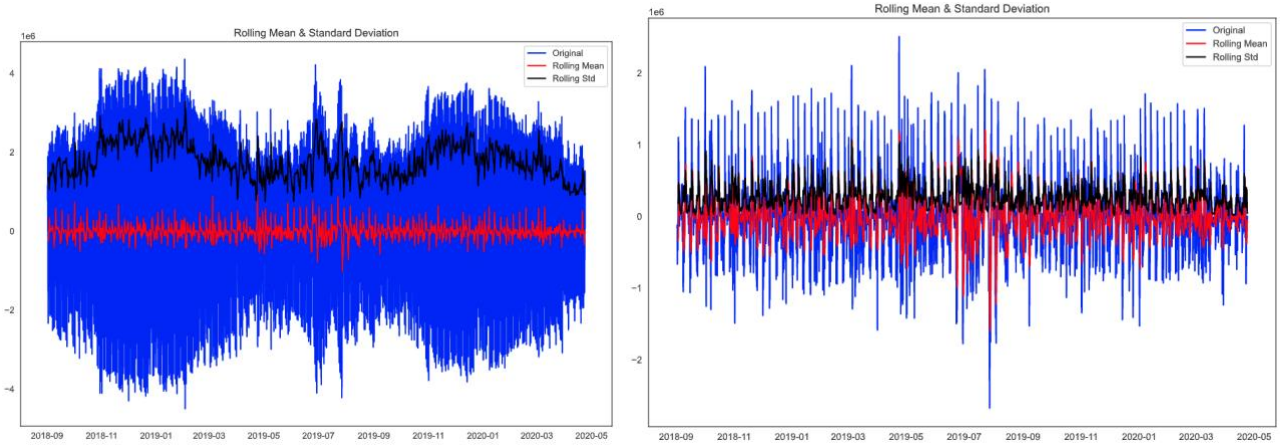


Figura 13: media e standard deviation prendendo in considerazione i valori con shift 12.
Figura 14: media e standard deviation prendendo in considerazione i valori con shift 24.

Per validare la tesi che la serie storica sia stazionaria, è possibile utilizzare il test di Dickey-Fuller, che mi permette di valutare se sussista un trend nelle variabili che renda la regressione spuria (quando non è rispettata l'assunzione che le osservazioni sono indipendenti ed identicamente distribuite).
Espongo qui di seguito i risultati, ottenendo la veridicità dell'ipotesi di stazionarietà:

```
Valori risultato:
(-5.925080543755828, 2.454543900941714e-07, 42, 14381, {'1%': -3.4308047992224897, '5%': -2.861741000946377, '10%': -2.566876988062525}, 373548.482116547)
------------------------
ADF statistic: -5.925081
p-value: 0.000000
Critical Values:
        1%: -3.431
        5%: -2.862
        10%: -2.567
Reject Ho - Time Series is Stationary
```

Figura 15: output test di dickey-fuller per verifica stazionarietà.

Inoltre, effettuo anche il test di Kwiatkowski- Phillips-Schmidt-Shin (KPSS), ottengo come output l'ipotesi che la serie storica non sia stazionaria:

```
KPSS Statistic: 8.234963053484819
p-value: 0.01
num lags: 42
Critial Values:
   10% : 0.347
   5% : 0.463
   2.5% : 0.574
   1% : 0.739
Result: The series is not stationary
```

Figura 16: output test di Kwiatkowski- Phillips-Schmidt-Shin (KPSS)

Osservando che la serie storica per il test di Dickey-Fuller risulta stazionaria, mentre per il test di Kwiatkowski- Phillips-Schmidt-Shin (KPSS) la serie storica risulta non stazionaria, deduco che la serie storica è stazionaria per differenza e che sarà opportuno applicare una differenza stagionale. Qui di seguito osservo i primi duemila valori del training, con diversi valori di differenza, per osservare le differenze:

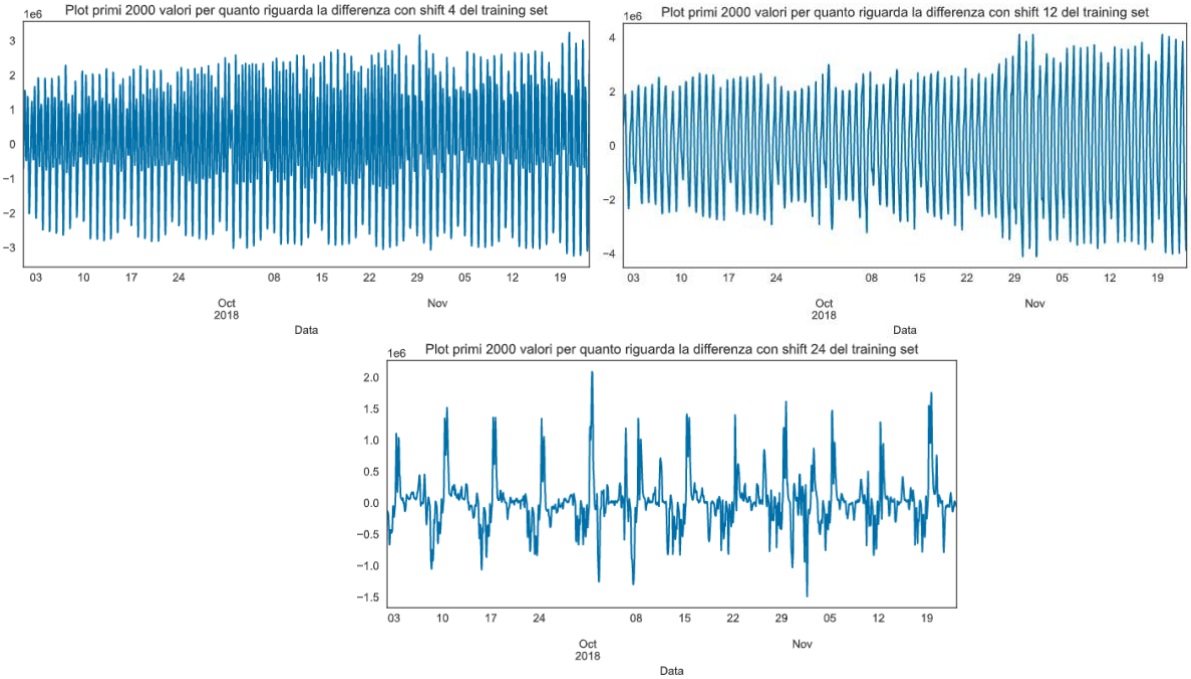


Figura 17: plot primi 2000 valori del training con shift = 4.
Figura 18: plot primi 2000 valori del training con shift = 12.
Figura 19: plot primi 2000 valori del training con shift = 24.

Rieseguo quindi i test di Dickey-Fuller e di Kwiatkowski- Phillips-Schmidt-Shin (KPSS). Osservo che a seguito dell'applicazione della differenza stagionale, entrambi i test affermano l'ipotesi che la serie storica sia stazionaria. Ecco i risultati degli output:

```
Valori risultato:
(-22.324515362942293, 0.0, 42, 14357, {'1%': -3.430805559627438, '5%': -2.861741336985628, '10%': -2.5668771669327635}, 373057.924978197)
------------------------
ADF statistic: -22.324515
p-value: 0.000000
Critical Values:
        1%: -3.431
        5%: -2.862
        10%: -2.567
Reject Ho - Time Series is Stationary
```

Figura 19 output test di dickey-fuller per verifica stazionarietà, con shift = 24.

```
KPSS Statistic: 0.0213671643845622
p-value: 0.1
num lags: 42
Critial Values:
   10% : 0.347
   5% : 0.463
   2.5% : 0.574
   1% : 0.739
Result: The series is stationary
```

Figura 20: output test di Kwiatkowski- Phillips-Schmidt-Shin (KPSS), con shift = 24.

Osservo quindi ACF e PACF per quanto riguarda sia i valori in forma originale, in formato con shift = 24 e in formato con shift = 24 e lag = 50:

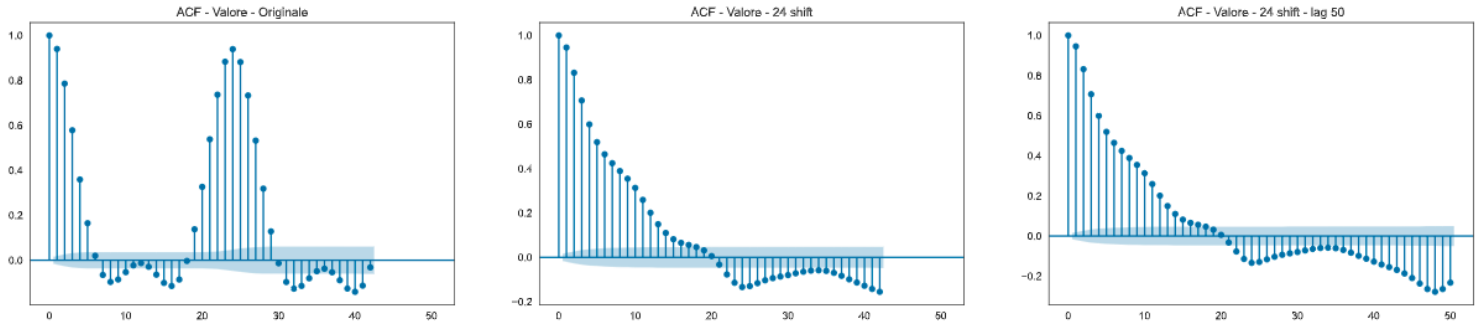


Figura 21: ACF con valori originali, con valori con shift = 24 e con valori shift = 24 e lag = 50.

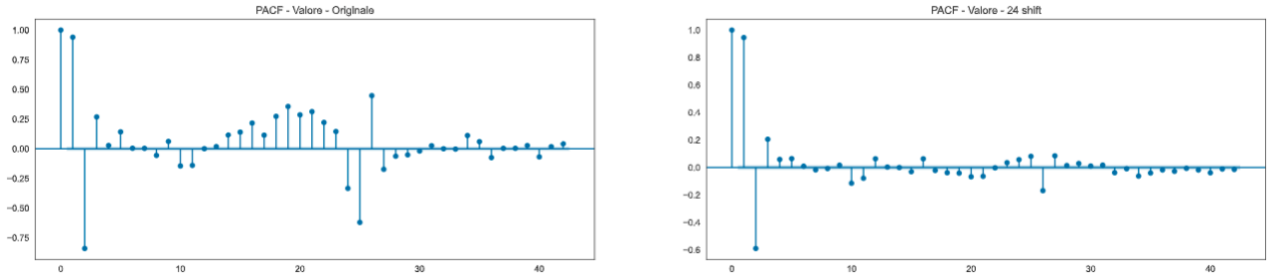


Figura 22: PACF con valori originali e, con valori con shift = 24.

Avendo quindi osservato la stazionarietà con differenza stagionale nei miei dati, posso passare ad effettuare i primi test di applicazione del modello.
A seguito delle varie combinazioni, posso tenere per P = 1, D = 1 e Q = 1, che rappresentano la parte stagionale del mio modello, andando a comporre il modello SARIMA, utilizzando un approccio di applicazione di varie combinazioni (grid search, non automatico, ma manuale per le visualizzazioni successive) di p, d e q, che rappresentano la parte non stagionale del mio modello, per poi effettuare un confronto visivo dei risultati del MAE tra training e validation set, cercando quindi di minimizzarlo il più possibile.
Viene riportato anche il valore dell'AIC (Akaike Information Criteria), che rappresenta una misura del modello statistico quantificata a partire dalla bontà di adattamento del modello e dalla complessità stessa del modello. Il modello che presenta il minore AIC di solito è il "migliore".

Combinazione (p,q)(P,Q, 24)	AIC	MAE Training Set	MAE Validation Set
SARIMAX(0,0,0)(1,1,1,24)	412513,969134	285325,014849	695463,831906
SARIMAX(0,0,1)(1,1,1,24)	401794,392292	163874,844687	697831,128300
SARIMAX(0,0,2)(1,1,1,24)	396150,195323	110570,021653	697338,905840
SARIMAX(2,0,0)(1,1,1,24)	373382,041967	67308,027093	688849,028113
SARIMAX(2,0,1)(1,1,1,24)	372992,051400	65868,885132	689606,974179
SARIMAX(2,0,2)(1,1,1,24)	372958,176543	65678,632932	689695,578857
SARIMAX(1,0,0)(1,1,1,24)	379067,622186	85184,168368	685561,576226
SARIMAX(1,0,1)(1,1,1,24)	374054,349905	69983,965473	688809,416923
SARIMAX(1,0,2)(1,1,1,24)	373165,407047	66552,748160	689660,573357

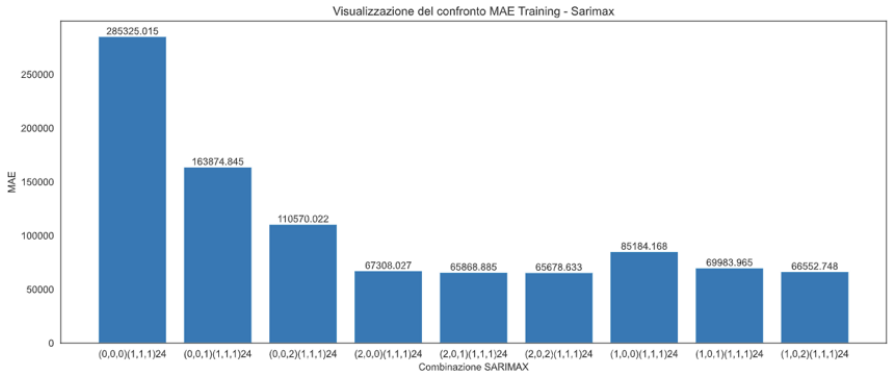


Figura 23: confronto risultati mae sul training set per le combinazioni di modelli sarimax.

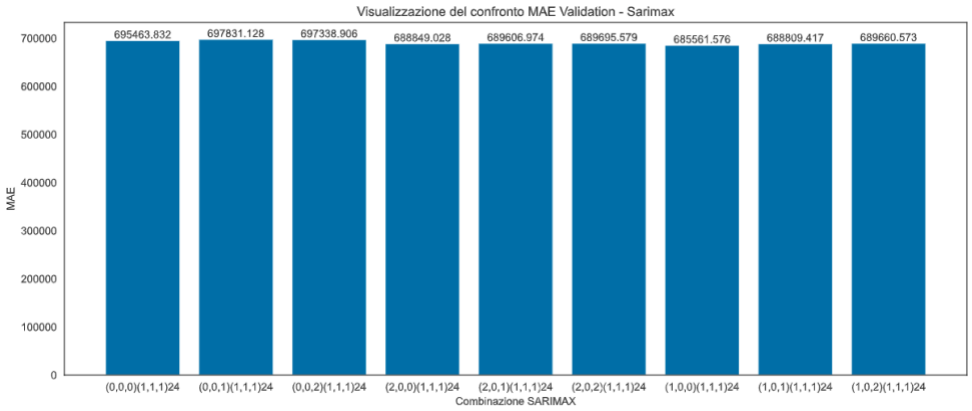


Figura 24: confronto dei valori di mae sul validation set per i modelli sarimax.

Ho deciso di valutare la forma logaritmica per aumentare il livello di stazionarietà della serie storica in considerazione. Ho applicato lo stesso approccio e processo precedente, ma prendendo la forma logaritmica del training e del validation set. Qui di seguito espongo i risultati:

Combinazione (p,q)(P,Q, 24) - log	AIC - log	MAE Training Set - log	MAE Validation Set - log
SARIMAX(0,0,0)(1,1,1,24)	-29292,911616	276194,731380	688041,443351
SARIMAX(0,0,1)(1,1,1,24)	-46049,860624	152804,020459	687804,941493
SARIMAX(0,0,2)(1,1,1,24)	-56549,135002	104836,063852	687503,991578
SARIMAX(1,0,0)(1,1,1,24)	-62600,044643	82885,339490	679285,454163
SARIMAX(1,0,1)(1,1,1,24)	-67724,454997	66755,408146	682798,489611
SARIMAX(1,0,2)(1,1,1,24)	-68603,367630	63261,950999	683518,542655
SARIMAX(2,0,0)(1,1,1,24)	-68199,807617	64515,850624	682329,115041
SARIMAX(2,0,1)(1,1,1,24)	-68612,628703	62786,022297	683677,524973
SARIMAX(2,0,2)(1,1,1,24)	-68685,072389	62577,781254	683904,206412

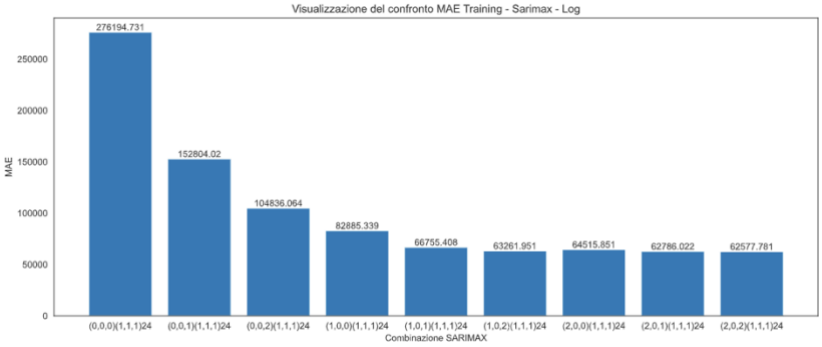


Figura 25: confronto risultati del mae sul training set per le combinazioni di modelli sarimax in forma logaritmica.

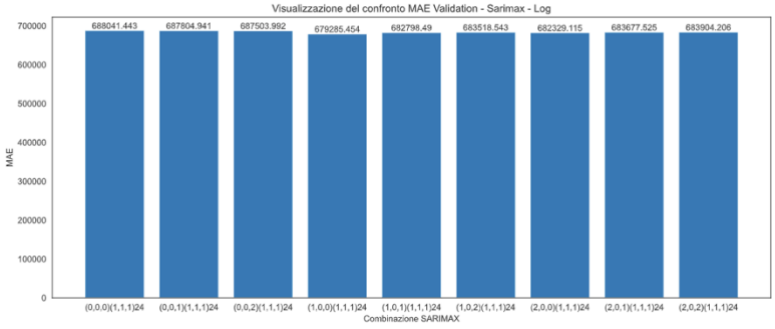


Figura 26: confronto dei valori di mae sul validation set per i modelli sarimax in forma logaritmica.

Osservo che applicare la trasformazione logaritmica, migliora leggermente i miei valori, osservando come il MAE del modello SARIMAX (2,0,2)(1,1,1,24) con trasformazione logaritmica sia il minore rispetto alle altre combinazioni in termini di AIC e per quanto riguarda il MAE, sebbene non sia il "migliore" come minimizzazione, risulta uno dei migliori. Decido quindi di utilizzare questa "configurazione" per effettuare il mio punto di partenza per l'ottimizzazione del modello SARIMA. Di seguito la previsione sul validation set con confronto con i dati effettivi:

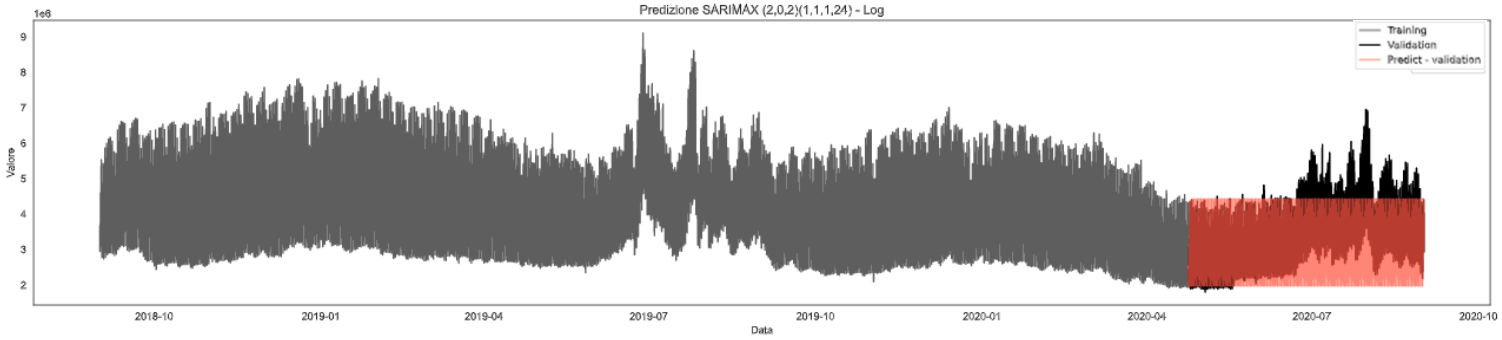


Figura 27: previsione dei dati di validation utilizzando il modello sarimax "migliore".

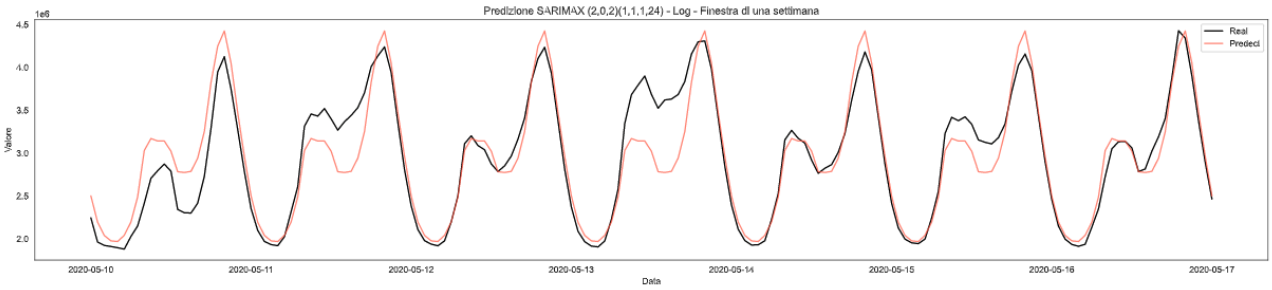


Figura 28: confronto della previsione del modello sarimax "migliore" in una finestra di una settimana.

Posso osservare come nel modello non venga presa in considerazione la stagionalità annuale e la stagionalità settimanale, che potremmo definire "weekend-settimanale" poiché da come abbiamo osservato i valori si mantengono abbastanza costanti durante i giorni lavorativi, per poi decrescere nel weekend.
Visualizzando la finestra di valori di una settimana, si osserva come la previsione rimanga sempre costante, quindi la stagionalità giornaliera sia stata rilevata. Procedo quindi con l'aggiunta di regressori esterni al modello SARIMA. Per prima cosa mi occupo di sistemare la stagionalità settimanale, confrontando i vari risultati tra i vari valori di armoniche che andrò ad utilizzare, successivamente andrò ad effettuare un *merge* tra le armoniche settimanali e le armoniche annuali, confrontando anche in questo caso i valori di MAE sul training e validation set rispetto alle varie combinazioni di armoniche, ecco di seguito i risultati (vengono riportati anche i valori di AIC, già esposto):

Armoniche settimanali	AIC	MAE – Training set	MAE – Validation set
20	-66779.420327	66057.688293	3312927.014668
11	-64358.788150	71117.086884	3210785.791062
10	-63221.782448	76563.919273	1038681.841557
7	-64522.439473	70917.194727	702108.825715
6	-69250.091219	62375.782491	679409.383103
5	-69189.453939	62678.371744	679112.775404
4	-69050.838241	62797.666528	684178.941138
3	-69010.555671	62508.986647	684780.714998
2	-68951.600115	62628.276557	684910.139423

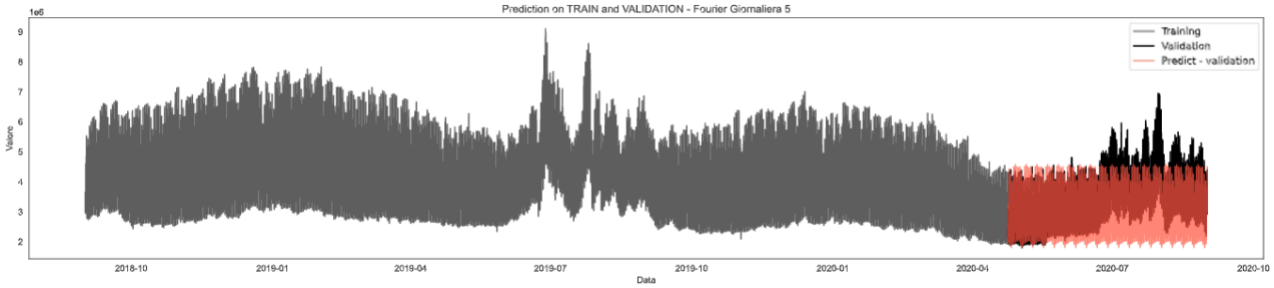


Figura 29: previsione del modello SARIMA avendo corretto la stagionalità settimanale con serie di Fourier con 5 armoniche settimanali.

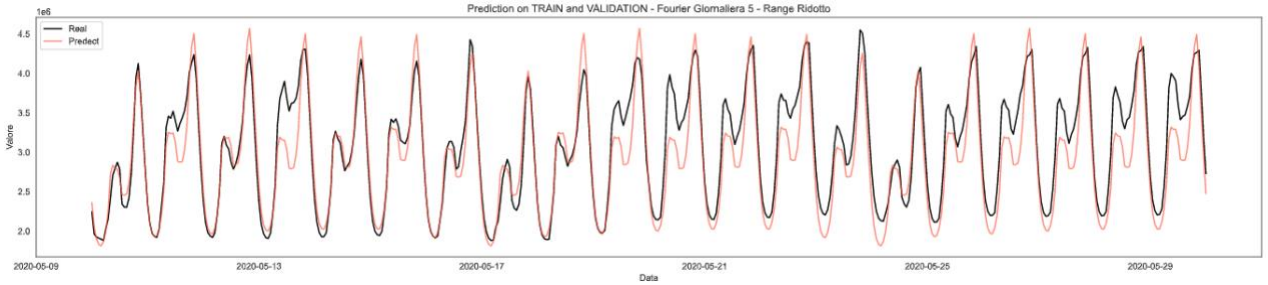


Figura 30: confronto della previsione del modello SARIMA "migliore" con stagionalità settimanale corretta in una finestra di una settimana con serie di Fourier con 5 armoniche settimanali.

Provvedo quindi ad effettuare l'unione con la serie di Fourier per correggere la stagionalità annuale:

Armoniche annuali – Armoniche settimanali	AIC	MAE – Training set	MAE – Validation set
2 – 5	-69228.008322	62600.201776	398330.384173
3-5	-69238.398910	62517.766086	346064.553343
4-5	-69238.356946	62534.483567	342706.205674
5-5	-69237.622214	62569.458733	375028.023167
7-5	-69241.407037	62575.142453	376851.286660
10-5	-69261.812679	62517.091155	380216.728881

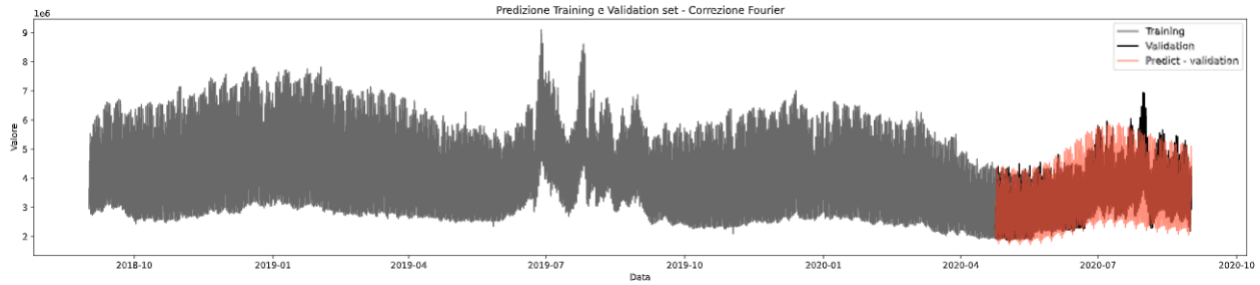


Figura 31: previsione del modello SARIMA avendo corretto la stagionalità settimanale e annuale con serie di Fourier con 5 armoniche settimanali e 3 armoniche annuali.

Osservano i risultati della serie di Fourier precedente, decido di utilizzare 5 armoniche per correggere la stagionalità settimanale ed utilizzare 4 armoniche per correggere la stagionalità annuale.
Avendo corretto le varie stagionalità riscontrate, posso passare ad effettuare la previsione del modello SARIMA sui miei dati, applicando quindi un modello SARIMAX (2,0,2)(1,1,1,24) con trasformazione logaritmica corretto con 5 armoniche per la stagionalità settimanale e 4 armoniche per correggere la stagionalità annuale.
Di seguito visualizzo la mia previsione sui dati futuri. Osservo che la stagionalità annuale sia stata rilevata correttamente dal modello.

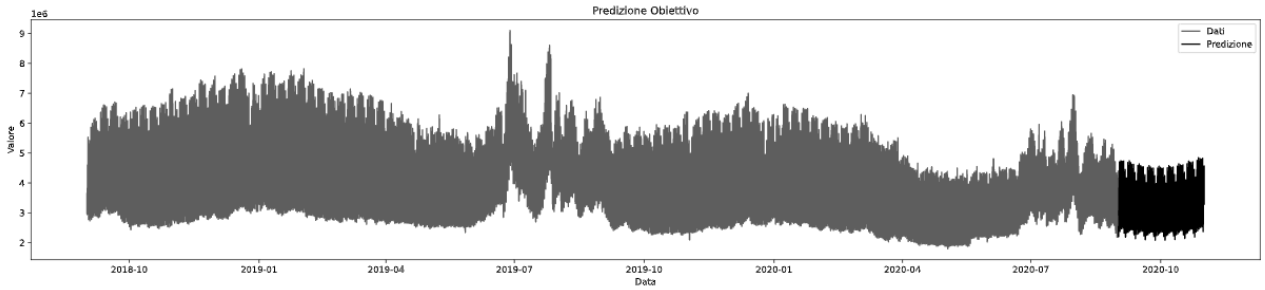


Figura 30: previsione del modello SARIMA definitivo per i modelli arima.

MODELLI UCM

I modelli Unobserved Components Model (UCM) performano una decomposizione della serie storica pensata come composta da componenti quali: trend, stagionalità e ciclo. Nella versione più classica ed estesa vi è anche la componente del white noise.
Il processo che ho attuato è stato quello di testare varie composizioni dei modelli UCM, osservando successivamente i valori di AIC, MAE sul training set e MAE sul validation set. Nella composizione di ogni modello ho indicato il periodo della componente stagionale pari a 24, ossia giornaliera,

inoltre, per modellare le componenti stagionali con funzioni trigonometriche ho indicato i periodi settimanali pari a 24*7 (168) con armoniche come serie di Fourier testando diversi valori. Riporto il nome completo dei modelli utilizzati:

- "ntrend" = No Trend;
- "dconstant" = Deterministic Constant;
- "llevel" = Local Level;
- "rwalk" = Random Walk;
- "dtrend" = Deterministic Trend;
- "lldtrend" = Local Linear Deterministic Trend;
- "rwdrift" = Random walk with drift;
- "lltrend" = Local linear Trend;
- "strend" = Smooth Trend;
- "rtrend" = Random Trend;

Dati la velocità computazionale nel calcolare I modelli UCM, ho verificato diverse situazioni di correzione delle armoniche per correggere la stagionalità settimanale, poiché voglio verificare come abbassando il numero di armoniche per correggere la stagionalità settimanale, riesco ad ottenere un leggero miglioramento del modello:

Tipologia UCM	Armoniche	AIC	MAE- Training Set	MAE – Validation Set
Rwalk	15	482350.860447	103060.012927	682731.952152
Dconstant	15	483252.476952	1027661.475699	3598309.558316
Ntrend	15	489778.861280	1027620.892037	3598369.484815
Llevel	15	483375.508269	104418.780408	682432.982220
Lldtrend	15	483342.126223	104393.585319	672887.582560
Rwdrift	15	482317.546417	103034.268613	673875.643769
Lltrend	15	484883.480352	105176.707607	11268159.194586
Strend	15	484877.743373	105179.157676	11270033.050383
Rtrend	15	483842.371721	104097.130613	11732591.260062
Dtrend	15	483058.621466	554155.159438	3692881.861085

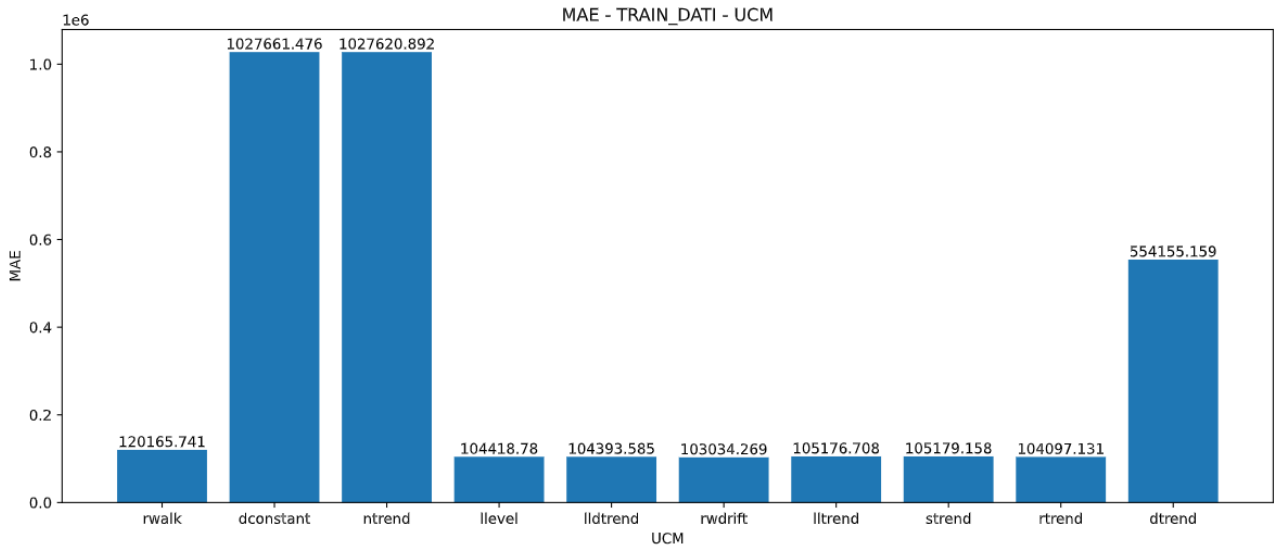


Figura 31: confronto MAE dei modelli UCM sul training set con 15 armoniche a correggere la stagionalità settimanale.

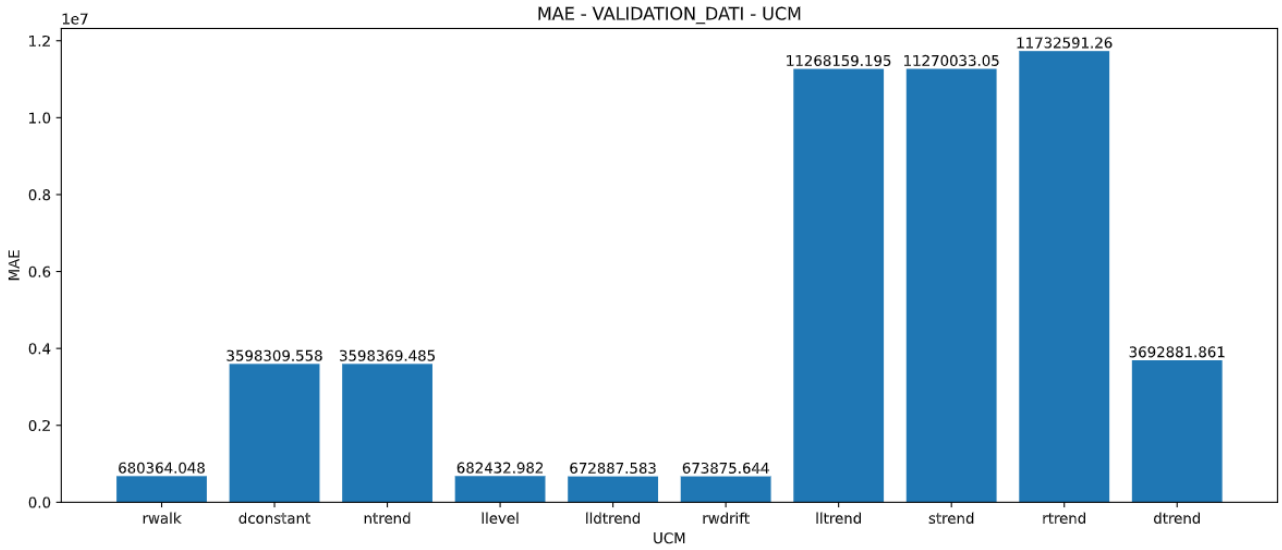


Figura 32: confronto MAE dei modelli UCM sul validation set con 15 armoniche a correggere la stagionalità settimanale.

Da una prima osservazione, il modello maggiormente performante risulta essere il lldtrend (Local Linear Deterministic Trend), sia in base al MAE sul validation set che in base al valore di AIC.
Eseguo il medesimo processo ma utilizzando 9 armoniche di Fourier per correggere la stagionalità settimanale, ecco i risultati:

Tipologia UCM	Armoniche	AIC	MAE- Training Set	MAE – Validation Set
Rwalk	9	472180.639780	120165.740734	680364.048213
Dconstant	9	473471.400830	730705.775714	3600090.130798
Ntrend	9	479996.853387	730722.878124	3600151.767917
Llevel	9	473711.228213	123499.319441	680002.153841
Lldtrend	9	473678.500359	123440.915640	670466.444402
Rwdrift	9	472148.011133	120110.860721	671894.468286
Lltrend	9	475274.271767	126545.011174	1366961.206797
Strend	9	475265.857525	126549.334907	1361607.291688
Rtrend	9	473714.273995	123186.559896	547220.726858
Dtrend	9	473273.736168	401996.309978	3698432.474384

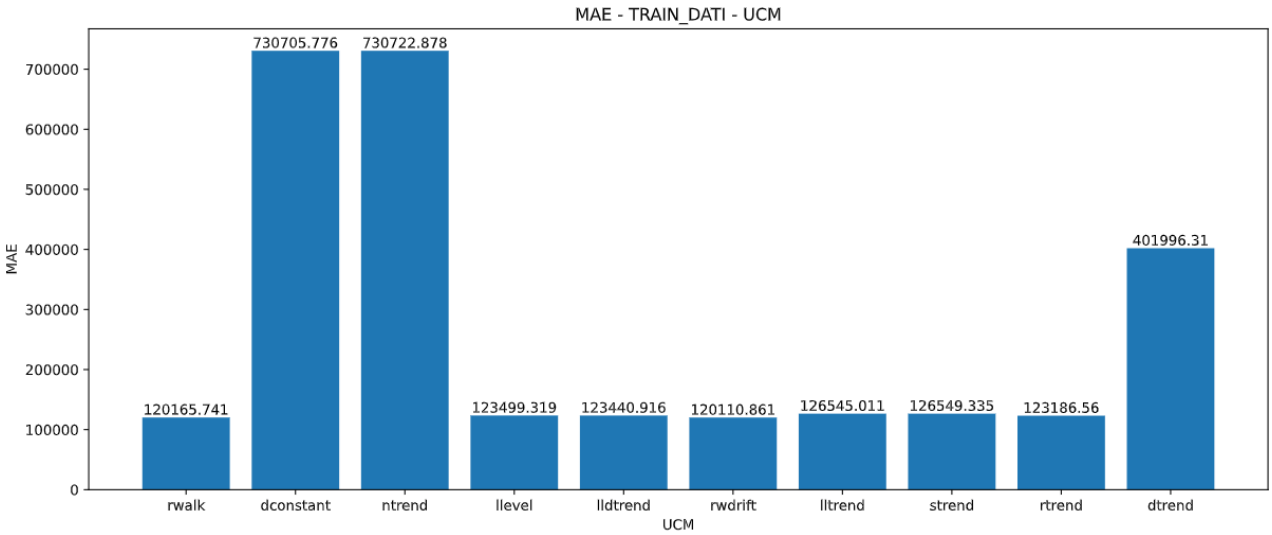


Figura 33: confronto MAE dei modelli UCM sul training set con 9 armoniche a correggere la stagionalità settimanale.

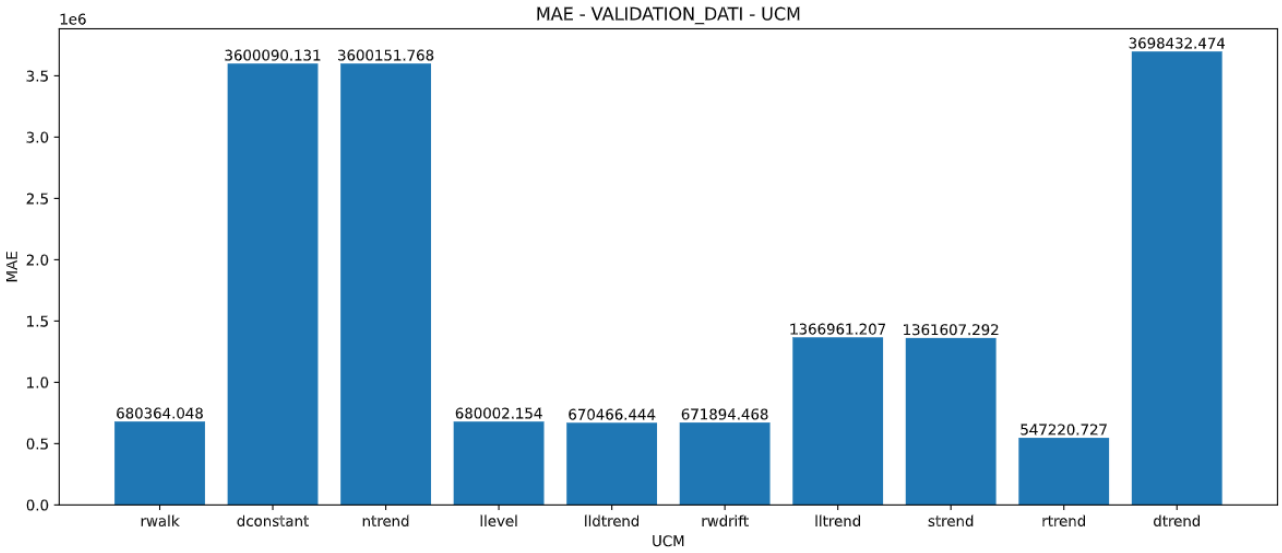


Figura 34: confronto MAE dei modelli UCM sul validation set con 9 armoniche a correggere la stagionalità settimanale.

Ulteriore verifica che il modello "lldtrend" sia la soluzione migliore come modello UCM è data dai modelli UCM con 9 armoniche per correggere la stagionalità settimanale. Per non effettuare il calcolo con varie armoniche per tutti le varie composizioni, proseguo solamente con il "lldtrend", calcolando sempre la stagionalità settimanale ma con varie armoniche, per osservare quale mi riduce maggiormente il MAE:

Modello	Armoniche	AIC	MAE- Training Set	MAE – Validation Set
Lldtrend	5	465877.185044	149994.794060	693957.304179
Lldtrend	6	472148.011133	120110.860721	671894.468286
Lldtrend	7	469976.897893	125660.183386	667073.544516
Lldtrend	8	471869.981785	124424.987496	674352.404625
Lldtrend	9	473678.500359	123440.915640	670466.444402
Lldtrend	15	483342.126223	104393.585319	672887.582560

Osservando i risultati, scelgo di proseguire con il "lldtrend" con 7 armoniche per correggere la stagionalità settimanale. Addestro il modello anche in formato logaritmico dei dati per capire se vi è la presenza di eventuali migliorie:

Modello	Armoniche	AIC	MAE – Training Set	MAE – Validation Set
Lldtrend – log	7	-56932.958565293	108406.534541	792897.550233

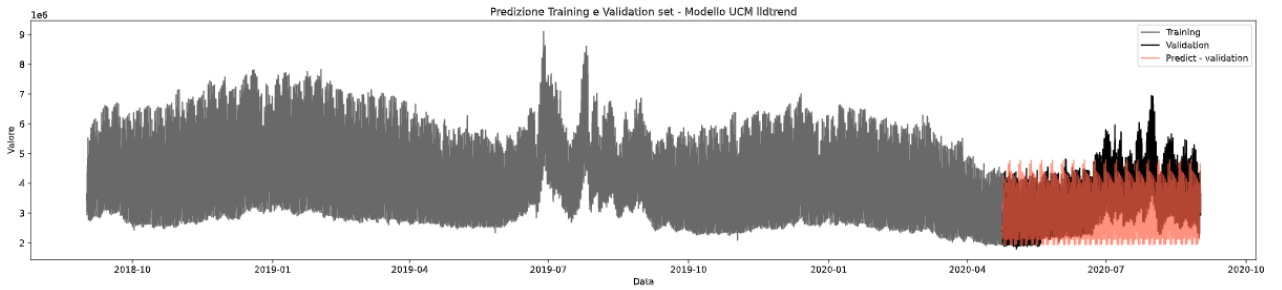


Figura 35: previsione del modello UCM "migliore" lldtrend con correzione stagionalità settimanale con 7 armoniche.

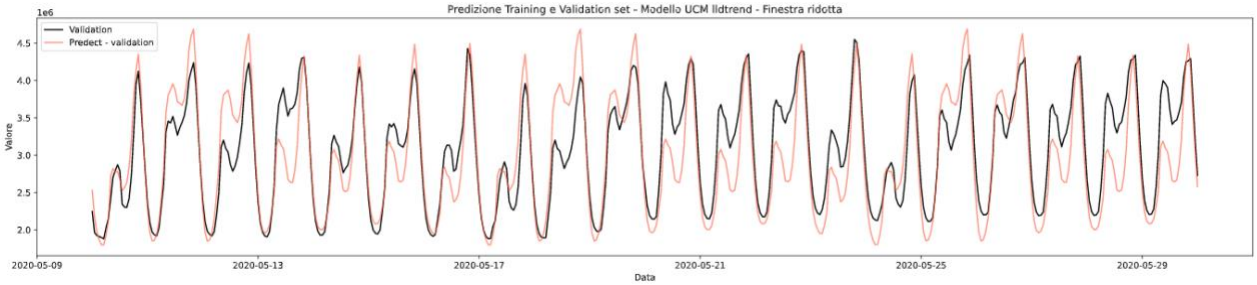


Figura 36: previsione del modello UCM "migliore" lldtrend con correzione stagionalità settimanale con 7 armoniche finestra di due settimane circa.

Come nel caso precedente dei modelli SARIMA, osservo come il modello abbia colto la stagionalità giornaliera e settimanale, ma per migliorarlo ulteriormente, è necessario correggere la stagionalità annuale.
Utilizzo una stagionalità annuale calcolando varie armoniche per correggere maggiormente la stagionalità della serie storica. Vado ad effettuare dei test aggiungendo la componente ciclo, per valutare la presenza di eventuali migliorie nel modello. Ecco i risultati delle varie prove:

Modello	Armoniche	AIC	MAE – Training Set	MAE – Validation Set
Lldtrend	7 giornaliera + 1 annuale + cycle = true	470823.208584	124307.444070	448588.534079
Lldtrend	7 giornaliera + 1 annuale + cycle = false	470886.372293	124383.407123	448588.534077
Lldtrend	7 giornaliera + 2 annuale + cycle = True	471619.563428	123565.389298	888231.278115
Lldtrend	7 giornaliera + 2 annuale + cycle = False	471682.842794	123628.467515	888231.278154
Lldtrend	7 giornaliera + 3 fourier annuale	469988.897893	125660.183375	667073.497244
Lldtrend	7 giornaliera + 5 fourier annuale	469996.897893	125660.183224	667073.198141

Scelgo come modello definitivo un modello UCM con lldtrend, correggendo la stagionalità settimanale con 7 armoniche e correggendo la stagionalità annuale con 1 armonica.
Ecco i risultati del plot della previsione:

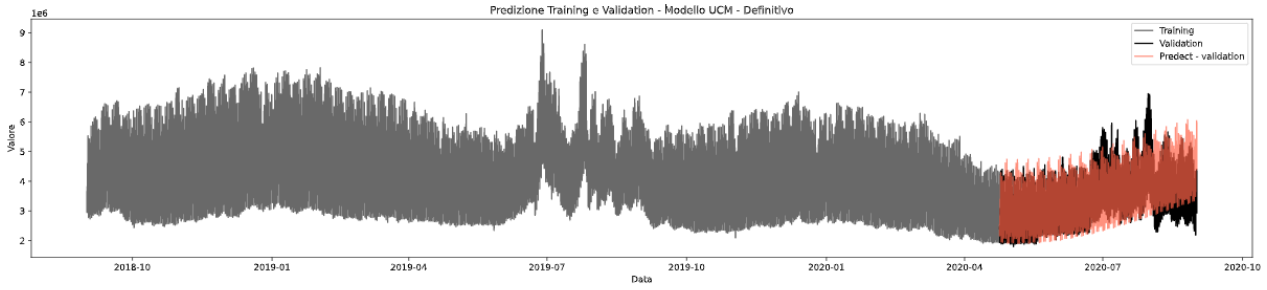


Figura 37: previsione del modello UCM "migliore" sui valori del validation

Effetto quindi la previsione con la seguente composizione del modello UCM:

- Modello "lldtrend" (Local Linear Deterministic Trend);
- Stagionalità giornaliera;
- Stagionalità settimanale corretta con 7 armoniche di serie di Fourier;
- Componente ciclo;
- Stagionalità annuale corretta con 1 armonica di serie di Fourier;

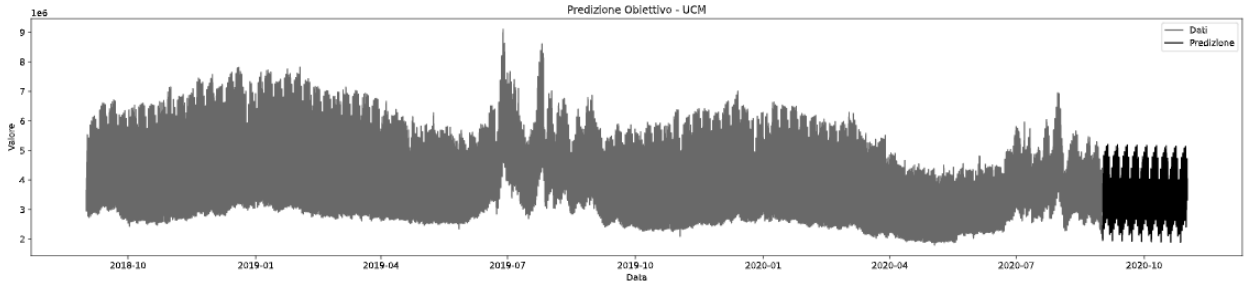


Figura 38: previsione del migliore modello UCM "lldtrend" sul periodo di dati futuri.

MODELLO PROPHET

Prophet è un applicativo per la previsione di serie storiche basata su un modello additivo dove i trends non-lineari sono addestrati con stagionalità giornaliera, settimanale ed annuale, dove è possibile inserire anche una stagionalità "festiva", indicando quindi i giorni di festività. È adatto a serie storiche che presentano delle forti stagionalità, come nel caso di studio. Per poter "leggere" i dati, Prophet necessita di rinominare le features con nome "ds" per le date e "y" per il dato numerico. Lo sviluppo di Prophet è avvenuto tramite Google Colab a causa di alcuni errori in locale nell'istallazione di alcune librerie.
Ho deciso di utilizzare un processo partendo dall'introduzione iniziale di pochi parametri, andando progressivamente ad aumentare la complessità del modello Prophet, avendo come metro di valutazione sempre il MAE sul validation set.
Di seguito espongo i componenti dei vari modelli utilizzati:

Componenti	Add.seasonality	MAE Training Set	MAE Validation Set
interval_width=0.95, yearly_seasonality=True, weekly_seasonality=True	(name='monthly', period=30.5, fourier_order=2, prior_scale=0.02)	363690.4	1068933.0
interval_width=0.95, yearly_seasonality=10, weekly_seasonality=3	(name='hourly', period=30.5, fourier_order=3, prior_scale=0.02)	363542.2	1070384.8
yearly_seasonality=True, weekly_seasonality=True, changepoint_prior_scale=0.01		367052.2	591879.7
yearly_seasonality= False, weekly_seasonality = False, daily_seasonality=False, mcmc_samples=30, changepoint_prior_scale=0.01, seasonality_prior_scale=0.01	(name='weekly', period=168, fourier_order=5, prior_scale=0.1) , (name='yearly', period=365.25, fourier_order=3, prior_scale=0.1) , (name='quarterly', period=365.25/4, fourier_order=5, prior_scale=0.1) , (name='daily', period=1, fourier_order=4, prior_scale=0.1) , (name='monthly', period=30.5, fourier_order=5, prior_scale=0.1), add_country_holidays(country_name='US')	398794.6	547079.6

Il modello ultimo presentato con le sue componenti è il modello 4, che riduce il MAE sul validation rispetto alle configurazioni precedenti. Di seguito riporto la previsione sul periodo del validation set:

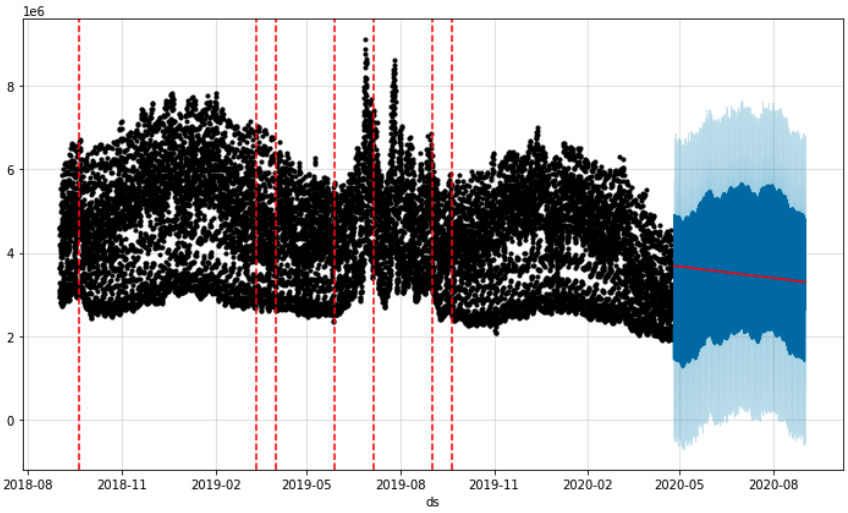
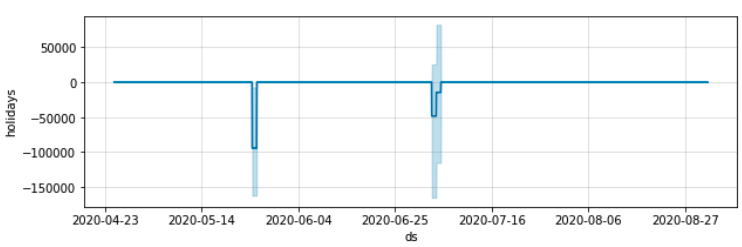
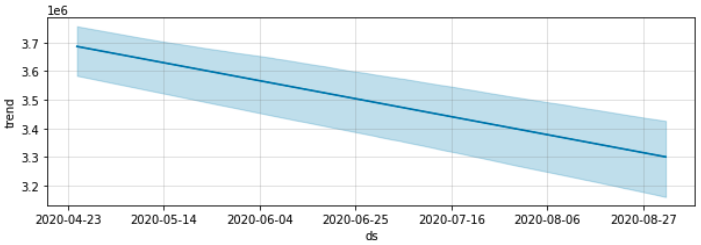
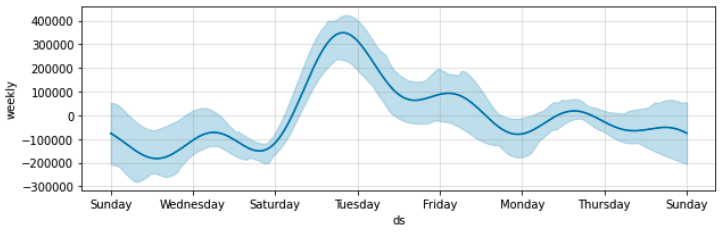
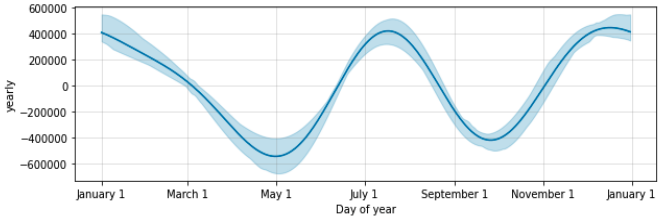


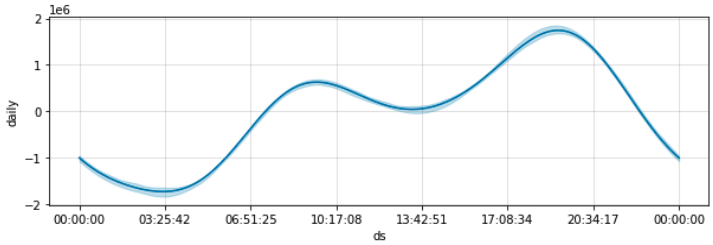
Figura 39: previsione del migliore modello PROPHET sul periodo del validation set.

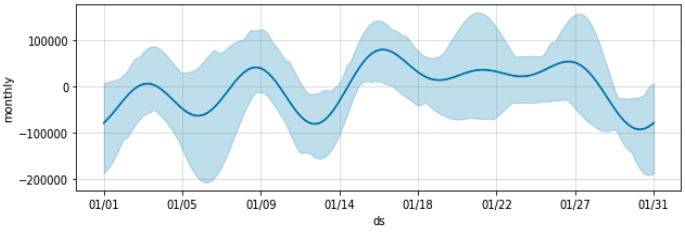












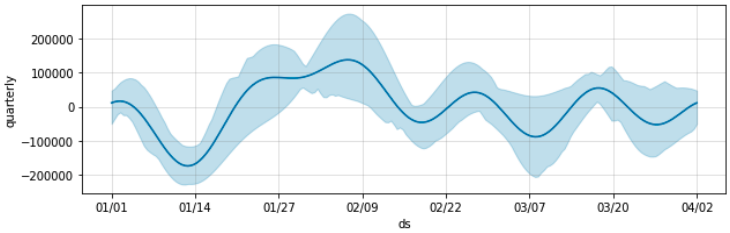


Figura 40: componenti del modello Prophet considerato "migliore".

Oltre alle componenti individuate precedentemente negli altri modelli ARIMA ed UCM, ho aggiungo anche come componenti stagionali "mensili" e "quadrimestrali" poiché ho osservato che introducendole il modello migliorasse in termini di MAE sul validation set, rispetto al non inserirle. Effetto la previsione con la modello 4 di Prophet:

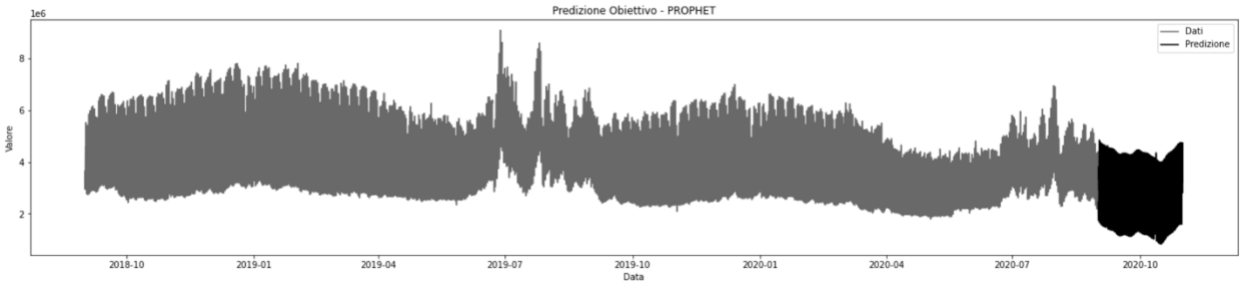


Figura 41: previsione del migliore modello PROPHET sul periodo di dati futuri.

MODELLI NON LINEARI – LSTM

Per quanto riguarda i modelli non-lineari, utilizzerò il framework LSTM (Long-short Term Memory), utilizzato per la predizione in ambito time series univariate. LSTM è un'architettura basata su Recurrent Neural Network (RNN) che è largamente utilizzata in ambito natural language processing e time series prediction.
Nella fase di preparazione dei dati, per renderli "accettabili" dal frameowork LSTM ho proceduto con il scalare i dati da 0 a 1, inoltre ho effettuato due split diversi, provando ad osservare i risultati con training e validation con range diversi:

- 1° split: training con 15144 osservazioni fino alla data 2020-05-23 23:00:00 e validation con 2400 osservazioni dalla data 2020-05-24 00:00:00
- 2° split: training con 15888 osservazioni fino alla data 2020-06-23 23:00:00 e validation con 1656 osservazioni dalla data 2020-05-24 00:00:00

La scelta degli split effettuati ha avuto le seguenti motivazioni: per quanto riguarda il 1° split è stato dato un mese in più come dati di training per addestrare il modello, pensando che essendo un modello non-lineare, più dati avesse a disposizione meglio sia, per quanto riguarda il 2° split ho voluto aumentare ulteriormente i dati di addestramento del modello.
Per le previsioni multi-step effettuate, ho deciso di applicare tre diverse strategie, che presenterò singolarmente.
Per tutti i modelli utilizzati, ho deciso di utilizzare l'implementazione stateful, che mi permette di conservare memoria tra i batch in una epoca di addestramento.

PREVISIONI DIRECT OUTPUT VECTOR

Le previsioni direct output vector con LSTM sono effettuate in un momento unico, sull'intero intervallo di previsione. Essendo un modello lento nell'addestramento ho scelto di utilizzare 200 neuroni, ma di scegliere 1 iterazione per addestrare il modello, con un'epoca ad iterazione. La lentezza di adozione del modello risulta essere uno scoglio di adozione, presentando comunque risultati non accettabili in termini di MAE sia sul training che sul validation

1° split (15144 osservazioni nel training set e 2400 osservazioni nel validation set)
Per quanto riguarda il primo split osservo che la previsione risulta accettabile per quanto riguarda il MAE, anche se risulta comunque piuttosto elevato. Rispetto ai risultati di split 2, si può osservare come la predizione risulta più spostata verso l'alto, ad un livello maggiormente ottimale rispetto appunto allo split 2:

- MAE Training Set: 6030662.4;
- MAE Validation Set: 884008.8;

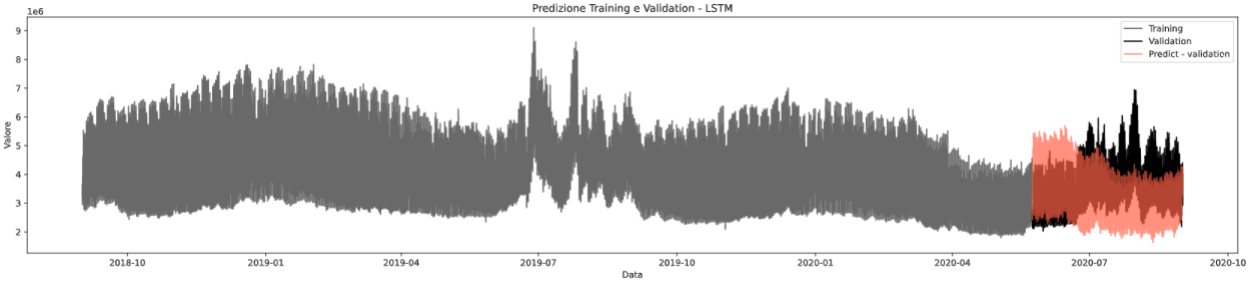


Figura 42: previsione 1° split con LSTM implementazione: Direct output vector.

2° split (15888 osservazioni nel training set e 1656 osservazioni nel validation set)
Rispetto allo split 1 il MAE sul validation set si alza, anche se presenta comunque un valore comunque accettabile, si potrebbe migliorare riducendo la quantità dei dati in input, vedendo che con meno dati comunque interpreta correttamente l'andamento.

- MAE Training Set: 7125007.5;
- MAE Validation Set: 927704.7;

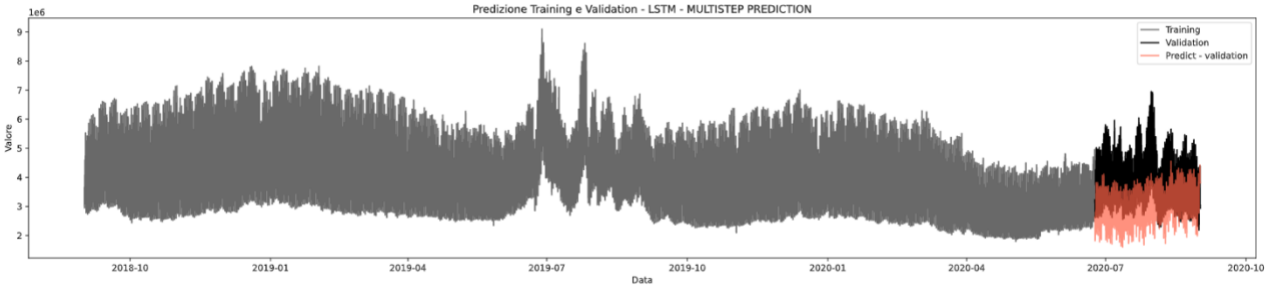


Figura 43: previsione 2° split con LSTM implementazione: Direct output vector

Nelle previsioni con metodo direct output vector, risulta migliore la previsione con split 1 dei dati (15144 osservazioni nel training set e 2400 osservazioni nel validation set).

PREVISIONI ONE-STEP AHEAD

Le previsioni One-Step ahead effettuano una previsione utilizzando in input i dati precedenti in modo ricorsivo. Per l'implementazione del modello ho deciso di utilizzare 200 neuroni, sarebbe stato ottimale effettuare un test con diversi valori di neuroni per capire come cambia il MAE finale, inoltre implemento soluzione stateful per la memoria dei dati di previsione. Il tempo di adozione di addestramento risulta ridursi per ogni step al crescere delle iterazioni ripetitive del modello, passando da 8 ms/step a 5 ms/step. Osservando i tempi di previsione relativamente brevi, ho deciso di effettuare in maniera ricorsiva 10 iterazioni con epoca 1, al di sopra di questo numero di iterazioni i tempi potrebbero diventare considerevoli, questa scelta è stata effettuata per entrambi gli split:

1° split (15144 osservazioni nel training set e 2400 osservazioni nel validation set)

Per quanto riguarda il primo split, posso osservare come le previsioni ed il MAE siano non accettabili e non "buoni":

- MAE Training Set: 11625942.0;
- MAE Validation Set: 1114480.7;

Si può osservare dalla rappresentazione grafica che la previsione rimane costante, senza cogliere l'andamento annuale della serie storica. Probabilmente dovuto all'arco temporale, aumentando il numero di osservazioni nel training ci si aspetta un miglioramento nel cogliere la stagionalità annuale, aspetto che cerco di osservare con lo split 2.

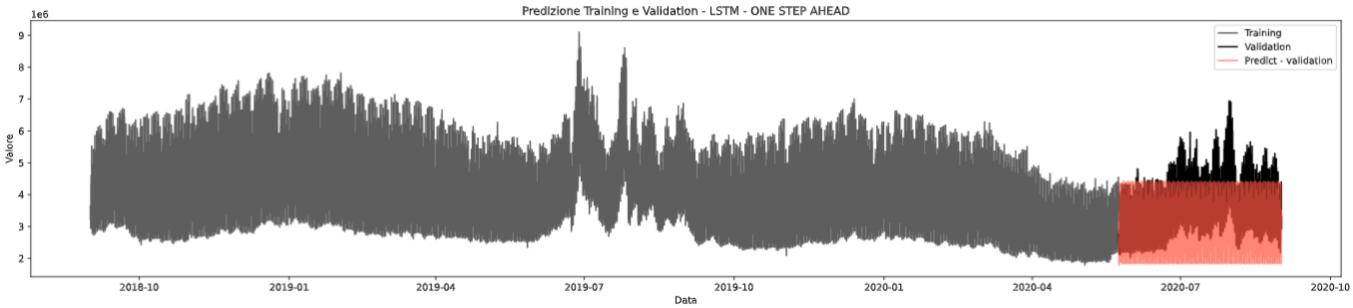


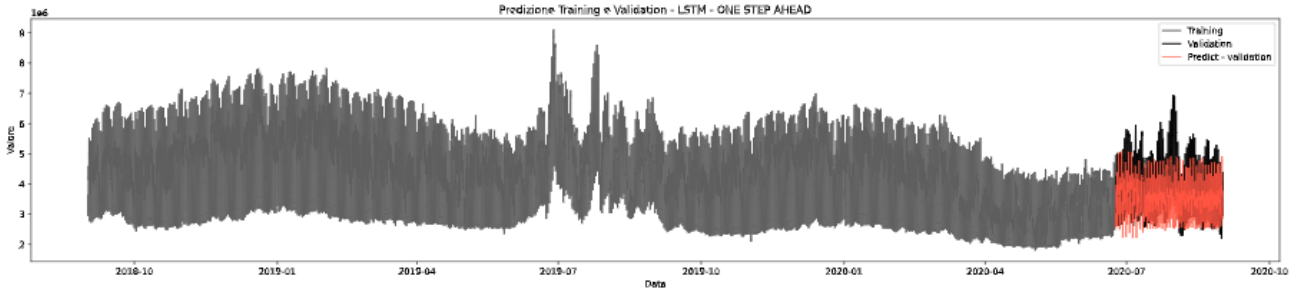
Figura 44: previsione 1° split con LSTM implementazione: One-step ahead

2° split (15888 osservazioni nel training set e 1656 osservazioni nel validation set)

Nello split precedente osservo che i risultati con questa metodologia non siano propriamente accettabili, quindi osservo con lo split 2, all'aumentare dei dati nell'addestramento se la situazione è anche solo leggermente migliorata:

- MAE Training Set: 14117657.1;
- MAE Validation Set: 917501.6;

Si può osservare come aumentando i dati di training, il modello migliori in termini di MAE sul training e validation set. Anche osservando la previsione (arancione in figura), la previsione stessa risulti costante come nell'implementazione precedente, ma cogliendo questa volta diversi livelli di



stagionalità.

Figura 45: previsione 2° split con LSTM implementazione: One-step ahead

Nelle previsioni con metodo one-step ahead, risulta migliore la previsione con split 1 dei dati (15144 osservazioni nel training set e 2400 osservazioni nel validation set).

PREVISIONI IBRIDE

Per quanto riguarda le previsioni ibride, vengono presi elementi da entrambe le implementazioni precedenti. In questa metodologia i risultati risultano essere maggiormente soddisfacenti, anche se i tempi di addestramento sono piuttosto lunghi, questo porta come nel caso delle previsioni direct

output vector a ridurre il numero di iterazioni sempre con epoca 1, poiché i tempi risulterebbero eccessivi. Anche in questo caso ho deciso di utilizzare 200 neuroni nel modello.

1° split (15144 osservazioni nel training set e 2400 osservazioni nel validation set)
Il valore di MAE sul validation set risulta abbastanza accettabile, ma decisamente superiore rispetto allo split 2, mostrando come lo split con più dati nella fase di addestramento sia una migliore soluzione per il modello.

- MAE Training Set: 7141710.7;
- MAE Validation Set: 917879.1

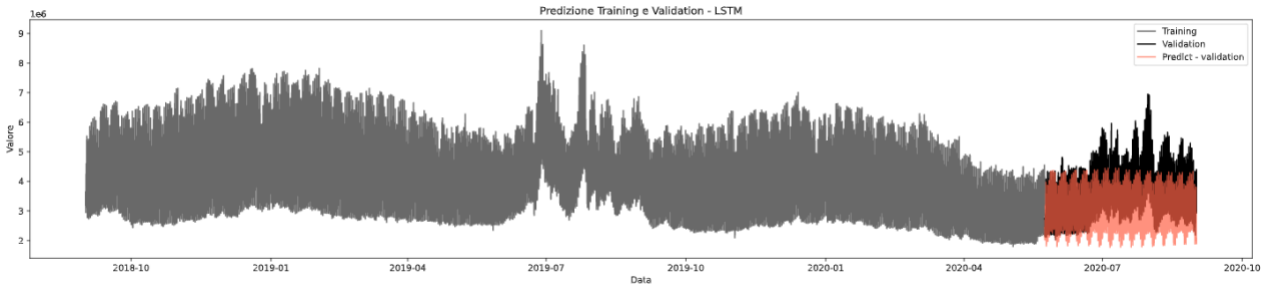


Figura 46: previsione 1° split con LSTM implementazione: ibrida.

2° split (15888 osservazioni nel training set e 1656 osservazioni nel validation set)
Si può osservare come il modello ibrido presenti dei risultati migliori rispetto alle metodologie precedenti.

- MAE Training Set: 3111967.8;
- MAE Validation Set: 706756.8;

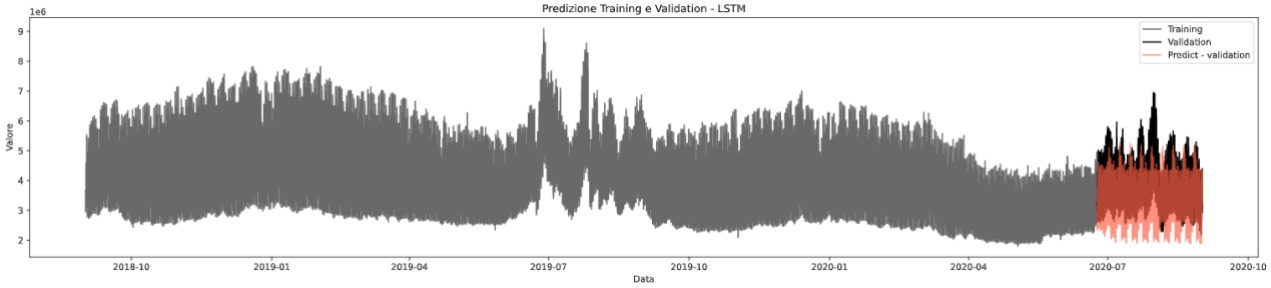


Figura 47: previsione 2° split con LSTM implementazione: ibrida.

Effettuo quindi la previsione finale, osservando come il MAE sul validation set del secondo split con il modello ibrido, abbia i risultati migliori e sicuramente più accettabili rispetto agli altri modelli. Ecco di seguito la previsione così effettuata:

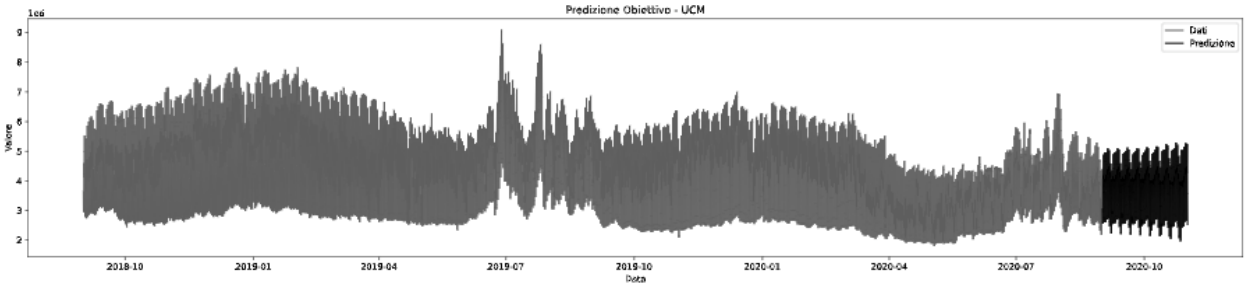


Figura 48: previsione con 2° split con LSTM implementazione: ibrida.

CONCLUSIONE

I modelli analizzati hanno permesso di stimare una previsione su un arco temporale futuro di 2 mesi, con una previsione senza essere a conoscenza dei dati effettivi. Volendo effettuare un ranking dei modelli migliori con le loro caratteristiche:

1) **Modello SARIMA**, MAE validation set 342706,205674;
2) **Modello UCM**, MAE validation set 448588,534079;
3) **Modello PROPHET**, MAE validation set 547079,6;
4) **Modello LSTM**, MAE validation set 706756,8.

Ranking	Modello	AIC	MAE – Validation set
1	SARIMA	-69238.356946	342706,205674
2	UCM	470823.208584	448588,534079
3	PROPHET	//	547079,600000
4	LSTM	//	706756,8

Per quanto riguarda i modelli lineari, nonostante il test su Prophet per comprendere meglio le stagionalità, il modello migliori in termini di mae risulta essere il modello SARIMA, seguito dal modello UCM e da Prophet. I modelli non-lineari non ottengono risultati accettabili, eccetto l'impostazione ibrida per LSTM, che presenta un MAE accettabile, sempre sul validation. Probabilmente, l'utilizzo di ulteriori implementazioni nei modelli non-lineari, come ad esempio CNN o un modello encoder-decoder, potrebbero portare a risultati migliori, inoltre la scelta dello split dei dati potrebbe essere da migliorare con ulteriori test con più o meno dati di training.

RISORSE UTILIZZATE

Riporto in maniera massiccia la documentazione e risorse utilizzate a vario titolo ed a diverso livello di dettaglio, per una maggiore trasparenza nel reperimento delle risorse.

https://www.dataquest.io/blog/python-datetime-tutorial/ ;
https://towardsdatascience.com/how-to-plot-time-series-86b5358197d6 ;
https://towardsdatascience.com/time-series-in-python-exponential-smoothing-and-arima-processes-2c67f2a52788 ;
https://www.dataquest.io/blog/tutorial-time-series-analysis-with-pandas/ ;
https://machinelearningmastery.com/gentle-introduction-autocorrelation-partial-autocorrelation/ ;
https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/ ;
https://robjhyndman.com/hyndsight/seasonal-periods/ ;
https://towardsdatascience.com/machine-learning-part-19-time-series-and-autoregressive-integrated-moving-average-model-arima-c1005347b0d7 ;
https://towardsdatascience.com/analyzing-seasonality-with-fourier-transforms-using-python-scipy-bb46945a23d3 ;
https://pythonawesome.com/pythons-best-automated-time-series-models/ (da guardare);
https://jakevdp.github.io/PythonDataScienceHandbook/03.11-working-with-time-series.html ;
https://kanoki.org/2020/04/30/time-series-analysis-and-forecasting-with-arima-python/ ;

https://www.statsmodels.org/devel/examples/notebooks/generated/statespace_sarimax_stata.html?highlight=sarimax
https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/
https://www.digitalocean.com/community/tutorials/a-guide-to-time-series-forecasting-with-arima-in-python-3 (risorsa molto utile anche con parecchie visualizzazioni)
https://www.wisdomgeek.com/development/machine-learning/sarima-forecast-seasonal-data-using-python/ (sarimax)
https://www.seanabu.com/2016/03/22/time-series-seasonal-ARIMA-model-in-python/
https://robjhyndman.com/hyndsight/dailydata/
https://stats.stackexchange.com/questions/215865/fourier-terms-to-model-seasonality-in-arima-models (risorse Fourier)
http://www.fis.unical.it/astroplasmi/primavera/dottorato/Analisi_di_Fourier.html
https://www.statsmodels.org/stable/examples/notebooks/generated/deterministics.html
https://fischerbach.medium.com/introduction-to-fourier-analysis-of-time-series-42151703524a
https://medium.com/intive-developers/forecasting-time-series-with-multiple-seasonalities-using-tbats-in-python-398a00ac0e8a (risorse fourier python)
https://tanzu.vmware.com/content/blog/forecasting-time-series-data-with-multiple-seasonal-periods
https://towardsdatascience.com/taking-seasonality-into-consideration-for-time-series-analysis-4e1f4fbb768f
https://towardsdatascience.com/time-series-forecasting-with-statistical-models-in-python-code-da457a46d68a
https://towardsdatascience.com/an-end-to-end-project-on-time-series-analysis-and-forecasting-with-python-4835e6bf050b
https://towardsdatascience.com/a-quick-start-of-time-series-forecasting-with-a-practical-example-using-fb-prophet-31c4447a2274
https://machinelearningmastery.com/time-series-forecasting-with-prophet-in-python/
https://towardsdatascience.com/time-series-forecasting-using-auto-arima-in-python-bb83e49210cd
https://www.researchgate.net/publication/341609757_Comparative_Analysis_of_Recurrent_Neural_Network_Architect ures_for_Reservoir_Inflow_Forecasting
https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/
https://www.kdnuggets.com/2018/11/keras-long-short-term-memory-lstm-model-predict-stock-prices.html