Streaming Data Management and Time Series Analysis

Testa Luca mat. 816000

***Abstract*—In questo lavoro vengono proposti una serie di modelli tra quelli lineari (ARIMA e UCM) e di machine learning (GRU e LSTM) volti ad affrontare un task di predizione di serie temporali. I modelli verranno valutati e confrontati sulla base della metrica MAE, e tramite una valutazione grafica. Sarà possibile quindi vedere come si comportano i diversi modelli con la serie storica data, portando ad alcune interessanti osservazioni.**

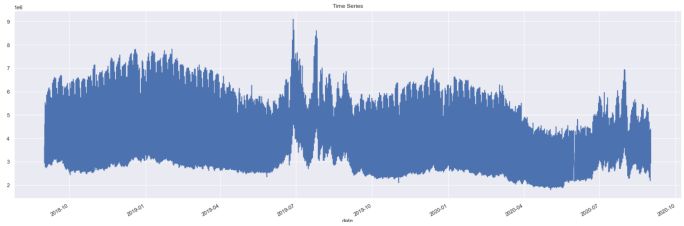

Figure 1. Time Series

I. Introduzione

Lo scopo del progetto è quello di testare diverse metodologie, tra cui un modello autoregressivo integrato a media mobile (ARIMA), un modello a componenti non osservabili (UCM) e una rete neurale ricorsiva (RNN). Si richiede una previsione nel periodo dal 1 Settembre 2020 al 31 Ottobre 2020. La serie (raffigurata in Fig. 1) rappresenta un'aggregazione oraria dei dati a disposizione. Il periodo di riferimento di questi dati va dal 1 Settembre 2018 al 31 Agosto 2020 per un totale di 17518 record.
La condizione di stazionarietà è verificata tramite Dickey-Fuller test, che rifiuta l'ipotesi nulla:

Test	Value
Test Statistics	-5.485109
P-Value	0.000002

Per la divisione del training set e test set è stato deciso per tutti i modelli di tenere dal 1 Aprile in poi per il test set, in maniera tale da tenere in considerazione l'impatto del Covid-19. Si è cosi ottenuta una proporzione rispettivamente del 80% e del 20%. Per la valutazione delle metodologie applicate, e per poter applicare in seguito un loro confronto, è stato scelta la metrica MAE. Inoltre, verrà comunque esaminata la predizione anche dal punto di vista grafico: si cercherà dunque un giusto trade-off tra le due analisi.

II. ARIMA

Si esegue inizialmente un'analisi tramite autocorrelation function (ACF) e partial autocorrelation function (PACF):

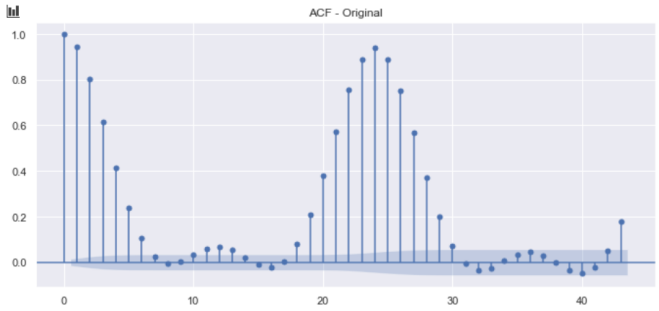

Figure 2. ACF

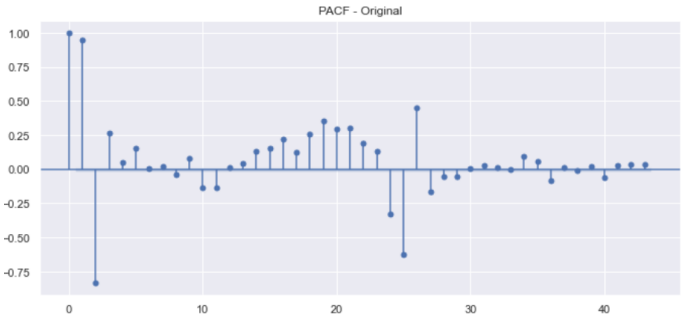

Figure 3. PACF

dove si vede chiaramente una stagionalità a 24. Si studia quindi un modello

$$AR(1)_{24}I(1)_{24}MA(1)_{24}$$

andando ad analizzare le componenti non stagionali probabilmente presenti. Per determinare dunque i parametri p, q dell'ARIMA $(AR(p)I(0)MA(q))$ si esegue tramite Grid Search la ricerca del modello che massimizzi la'AIC, con parametri p = 0,...5 e q = 0,...3. Si trova così il modello migliore, ossia un

$$ARIMA(5,0,3)(1,1,1)_{24}$$

dove l'ACF e la PACF risultano adeguatamente modellati:

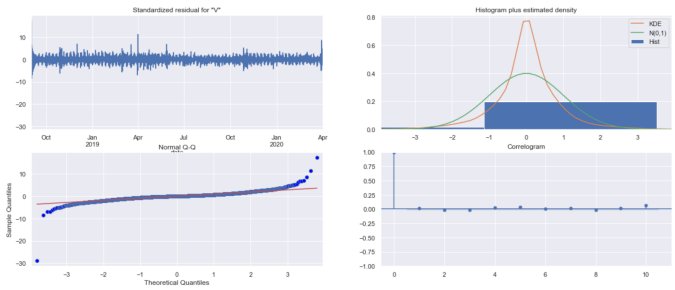

Figure 4. Arima 1

Vengono inoltre provati dei modelli con dei regressori esterni, nello specifico quattro diverse matrici di regressori:

- Weekend e Festività Italiane
- Covid-19
- FREQ, sin e cos con frequenza settimanale e annuale
- L'insieme delle tre sopra citate

Dopo una serie di test il modello migliore risulta essere:

$$ARIMA(5,0,3)(1,1,1)_{24}$$

$$[is\_weekend,\ is\_covid,\ sin365,\ cos365]$$

con un MAE TEST di *511'424*.

Covariance Type: opg

	coef	std err	z	P>|z|	[0.025	0.975]
is_weekend	-3.773e+05	3214.325	-117.371	0.000	-3.84e+05	-3.71e+05
is_covid	-1.281e+05	1.562	-8.2e+04	0.000	-1.28e+05	-1.28e+05
sin365	-8.547e+04	53.630	-1593.646	0.000	-8.56e+04	-8.54e+04
cos365	3.333e+05	30.351	1.1e+04	0.000	3.33e+05	3.33e+05
ar.L1	1.8116	0.026	70.219	0.000	1.761	1.862
ar.L2	-1.7255	0.032	-54.582	0.000	-1.787	-1.664
ar.L3	1.8024	0.033	54.883	0.000	1.738	1.867
ar.L4	-1.3629	0.031	-44.295	0.000	-1.423	-1.303
ar.L5	0.4412	0.011	40.332	0.000	0.420	0.463
ma.L1	-0.4289	0.026	-16.358	0.000	-0.480	-0.378
ma.L2	0.6502	0.013	51.018	0.000	0.625	0.675
ma.L3	-0.7253	0.025	-28.622	0.000	-0.775	-0.676
ar.S.L24	0.2924	0.007	42.942	0.000	0.279	0.306
ma.S.L24	-0.8675	0.005	-164.563	0.000	-0.878	-0.857
sigma2	1.667e+10	0.040	4.2e+11	0.000	1.67e+10	1.67e+10

Figure 5. Coefficienti

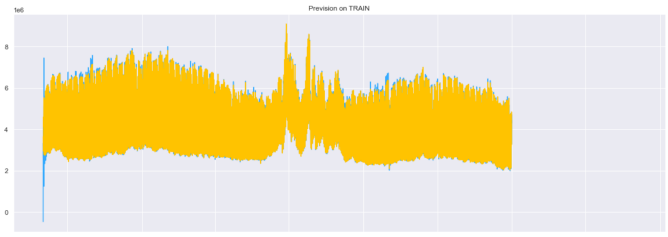

Figure 6. Previsione Test

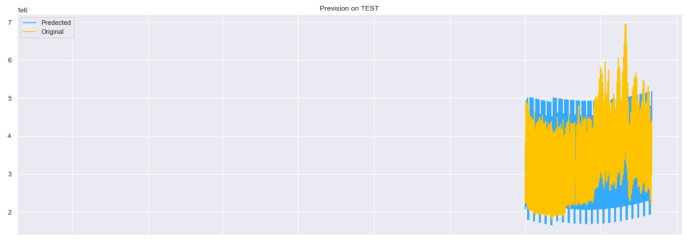

Figure 7. Previsione Finale

III. UCM

Anche in questo caso si è deciso di trovare la componente trend migliore sempre tramite approccio Grid Search, vedendo quale modello ottenesse MAE più bassi (train e test); è stato così selezionato il modello che prevedeva un livello con Random Walk con drift. Dopo di che si è deciso di testare i due regressori non sinusoidali per testarne la significatività: *Weekend e Covid*. Entrambi sono risultati significati ed inseriti all'interno del modello composto da un ciclo ed una stagionalità di 24 ore dummy. In seguito sono stati fatti numerosi test sul MAE con diversi modelli di stagionalità trigonometrica, una Grid Search sulle armoniche:

- RWDrift, ciclo, stagionalità dummy (24), due stagionalità trigonometrice (24*7 : 6 armoniche; 24*365 : 6 armoniche).
- RWDrift, ciclo, stagionalità dummy (24), due stagionalità trigonometrice (24*7 : 8 armoniche; 24*365 : 8 armoniche).
- RWDrift, ciclo, stagionalità dummy (24), due stagionalità trigonometrice (24*7 : 10 armoniche; 24*365 : 10 armoniche).
- RWDrift, ciclo, stagionalità dummy (24), due stagionalità trigonometrice (24*7 : 12 armoniche; 24*365 : 12 armoniche).
- RWDrift, ciclo, stagionalità dummy (24), due stagionalità trigonometrice (24*7 : 14 armoniche; 24*365 : 14 armoniche).

In questo caso il modello migliore è risultato essere l'ultimo descritto nella lista con 14 armoniche su doppia stagionalità trigonometrica e una dummy. Il tutto con MAE TEST di *1'045'537*.

ARMONICHE: 14

Unobserved Components Results

Dep. Variable:	VALORE	No. Observations:	13872
Model:	random walk with drift + stochastic seasonal(24) + stochastic freq_seasonal(168(14)) + stochastic freq_seasonal(8760(14)) + cycle	Log Likelihood	-233609.156
		AIC	467232.311
		BIC	467285.033
		HQIC	467249.876
Date:	Wed, 27 Jan 2021		
Time:	18:46:58		
Sample:	09-01-2018 - 04-01-2020		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
sigma2.level	3.438e+11	1.56e-16	2.2e+27	0.000	3.44e+11	3.44e+11
sigma2.seasonal	1.035e+12	5.28e-16	1.96e+27	0.000	1.03e+12	1.03e+12
sigma2.freq_seasonal_168(14)	1.035e+12	1.25e-15	8.25e+26	0.000	1.03e+12	1.03e+12
sigma2.freq_seasonal_8760(14)	1.035e+12	8.08e-16	1.28e+27	0.000	1.03e+12	1.03e+12
frequency.cycle	1.5708	0.405	3.879	0.000	0.777	2.364
beta.is_weekend	-2.269e+04	1.39e-10	-1.63e+14	0.000	-2.27e+04	-2.27e+04
beta.is_covid	-2982.8737	1.25e-12	-2.38e+15	0.000	-2982.874	-2982.874

Ljung-Box (L1) (Q):	1707.16	Jarque-Bera (JB):	5403.61
Prob(Q):	0.00	Prob(JB):	0.00
Heteroskedasticity (H):	0.69	Skew:	0.11
Prob(H) (two-sided):	0.00	Kurtosis:	6.06

Figure 8. Coefficienti

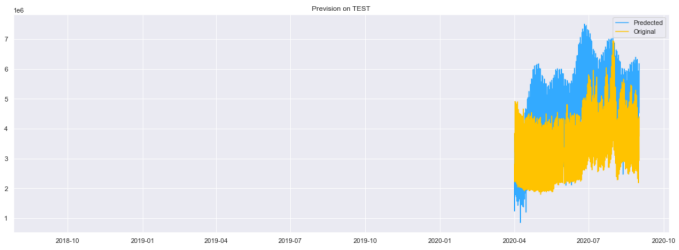

Figure 9. Previsione Test

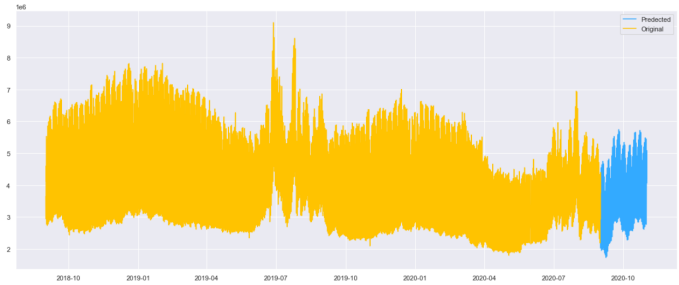

Figure 10. Previsione Finale

IV. RNN

Per valutare le performance di un modello di machine learning, e nello speciFIco mediante l'utilizzo di una RNN, è stato deciso di testare due architetture: una GRU e una LSTM. Per il pre-processing dei dati è stato utilizzato il MinMaxScaler, per poter scalare i dati un un range [0, 1]. Per la creazione della time series da inserire nel modello per l'addestramento, si è utilizzata una finestra che guarda indietro (chiamata look back) di 150 giorni. Dopodichè si è proceduto con la realizzazione dell'architettura, cosi composta:

- **GRU** o **LSTM** layer da 512 neuroni con attivazione *relu*
- Dropout di 0.33
- BatchNormalization
- Layer **Dense** con 512 neuroni e attivazione *relu*
- Dropout di 0.33
- Layer **Dense** con 512 neuroni e attivazione *relu*
- Dropout di 0.33
- Outpur **Dense** e attivazione *sigmoid*

Per la compilazione del modello si è usata come loss mae e come ottimizzatore Adam. Sono stati usati layer basati su GPU, del pacchetto keras; questo ha permesso una velocità di addestramento di circa 500 volte più veloce rispetto ai layer canonici. Si è stata impostata una batch size pari a 512, e un numero di epoche pari a 100. Sono state così confrontate le due architetture, che ha portato a scegliere il modello GRU in quanto riesce ad ottenere valori migliori con un numero di parametri e complessità minore rispetto a LSTM.
Si è ottenuto così un MAE TEST di *978'158*.

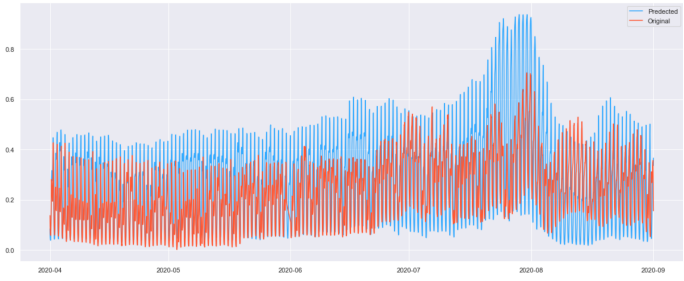

Figure 11. Previsione Test

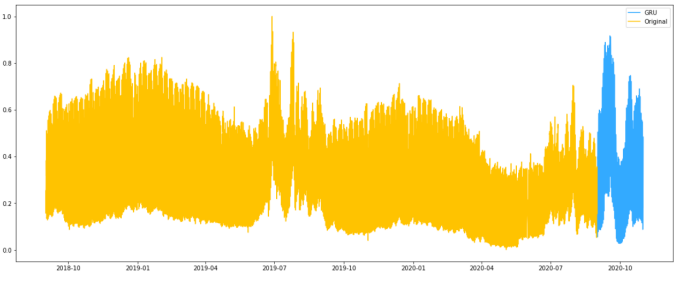

Figure 12. Previsione Finale

V. Conclusioni

Per concludere, andando a valutare i tre modelli possiamo vedere come, a livello di MAE, l'Arima sembri il migliore. Anche se, a livello grafico, UCM e RNN sembrano interpretare decisamente meglio i dati:

Modello	Valore	Run Time
Arima	511'424	11 Minuti
UCM	1'045'547	48 Secondi
RNN	978'158	15 Secondi

Si è voluto anche mostrare le differenze di run time tra i tre modelli, si nota come le rnn, utilizzando un enviroment basato su gpu, siano più veloci degli altri due.

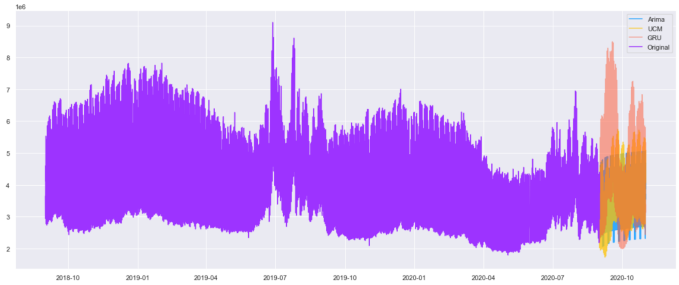

Figure 13. Previsione Finale di tutti i modelli

Si nota come tendenzialmente il modello GRU sembrerebbe seguire più l'andamento globale della serie (appunto perchè l'ha memorizzata per minimizzare l'errore), mentre il modello UCM segue di più l'andamento mensile. Infine, il modello ARIMA sembra aver colto bene la stagionalità settimanale, ma meno quella annuale.

Figure 14. Previsione Finale di tutti i modelli