Università degli studi di
Milano-Bicocca

Streaming Data Management and Time Series
Final project

Previsione time series univariata

Authors:
Ranieri Silvia, 878067, s.ranieri7@campus.unimib.it

A.Y.: 2021/2022

Contents

1 Analisi esplorativa **1**

2 Dati mancanti **1**

3 Train e Test **2**

4 Arima **2**
4.1 Stazionarietà 2
4.2 ACF e PACF 2
4.3 Parametri 3
4.4 Analisi di fourier 4

5 UCM **4**

6 Machine Learning **4**
6.1 LSTM 4
6.2 Gru 5

7 Previsioni orarie dal 2005-03-01 al 2005-03-31 **6**

8 Conclusioni **6**

Abstract

Lo scopo di questo progetto è di predire i valori orari di ossido di carbonio (CO) della time series univariata, il periodo di riferimento è dal 2005-03-01 (hour=0) al 2005-03-31 (hour=23). Verranno utilizzati diversi algoritmi e all fine stabilire: uno della famiglia ARIMA, uno della famiglia UCM, e uno della famiglia Machine Learning. Come strumento è stato utilizzato python. Le performance dei modelli sono stati confrontati tramite il Mean Absolute Percentage Error (MAPE) e il Mean Absolute Error (MAE) .

1 Analisi esplorativa

Il dataset è composto da 8526 osservazioni e da 3 colonne:

- 'Date' stringa contenente la data della misurazione, in formato yyyy-mm-dd.
- 'Hour' orario della misurazione in valori interi da 0 a 23.
- 'CO', contenente i valori relativi alla misurazioni di ossido di carbonio.

I dati a disposizione partono dal 10 marzo 2004 alle ore 18:00 fino al 28 febbraio 2005 ore 23:00. Inoltre non sono presenti cambi di ora (legale/solare).

2 Dati mancanti

Il primo passo è stato formattare la data e ora, attraverso il formato timestamp. Dall'analisi è emerso la mancanza di (365) valori relativi a CO circa 4.28% delle osservazioni totali. Inoltre si è notato una variazione oraria significativa con un trend giornaliero costante o molto simile, per cui si è deciso di rimpiazzare il valore con la media di 7 giorni prima e 7 giorni dopo della stessa ora. La serie temporale si presenta nel seguente modo:

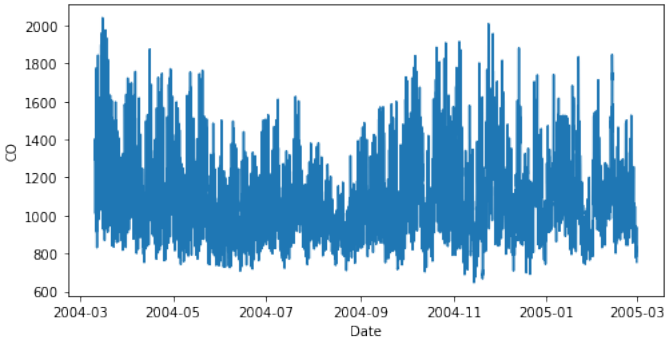

Figure 1: Serie storica

Notiamo delle variazioni in diminuzione soprattutto nei mesi estivi, infatti da un' analisi più accurata è emerso che vi sono dei picchi nei mesi di giugno, luglio e agosto. Per quanto riguarda gli orari si verifica una diminuzione nelle fasce dalle 5 alle 14. Dall'analisi del consumo medio settimanale possiamo osservare che il consumo aumenta nei giorni come martedì, mercoledì e giovedì. Inoltre vi è un aumento di consumo tra un anno e l'altro.

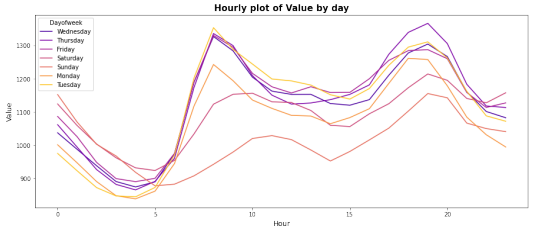

Figure 2: Analisi settimanale

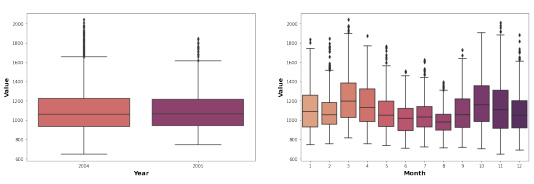

Figure 3: Analisi mensile

3 Train e Test

Successivamente si è passato alla suddivisione dei dati in train 80% e 20% test per confrontare l'andamento dei modelli. Inseguito si è passato allo studio della struttura della serie storica attraverso la decomposizione della serie, come si può notare dalla figura le varie componenti della serie.

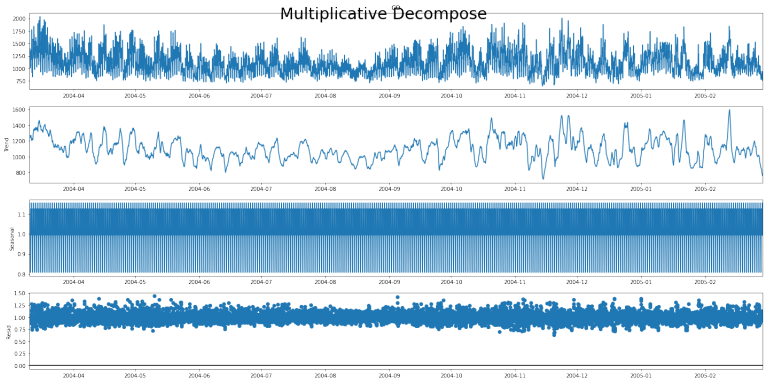

Figure 4: decomposizione della serie

4 Arima

4.1 Stazionarietà

Requisito fondamentale dei modelli ARIMA è la stazionarietà. (Trasformazioni come i logaritmi possono aiutare a stabilizzare la varianza di una serie temporale. Le differenze invece possono aiutare a stabilizzare la media di una serie temporale). Per verificare la presenza si è utilizzato due metodi: il test Augmented Dickey-Fuller e il test di Ljung-Box, il Kwiatkowski-Phillips-Schmidt-Shin (KPSS).

- **Dickey-Fuller**: p-value è di 4.113916e-12 la serie risulta stazionaria
- **Ljung-Box**: p-value 0.029522 serie non stazionaria

la serie è non stazionaria secondo il test KPSS a un livello di significatività del 0.95, perciò è necessario applicare la differenzazione. Sono stati testati modelli ARIMA con svariate combinazioni di parametri e svariate tipologie di regressori.

4.2 ACF e PACF

I modelli ARIMA utilizzano la funzione di autocorrelazione (ACF) e i grafici di autocorrelazione parziale (PACF) per determinare il numero di termini AR e / o MA. La figura mostra che l'ACF si assottiglia gradualmente fino a zero, mentre il PACF sembra avere un picco significativo.Questo ci fa desumere che potrebbe essere un buon ordine p=1 o p=2, P=1, Q=1. .

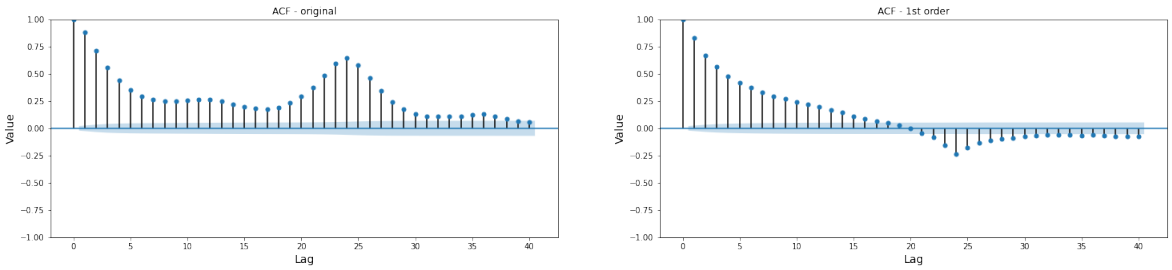

Figure 5: ACF della serie

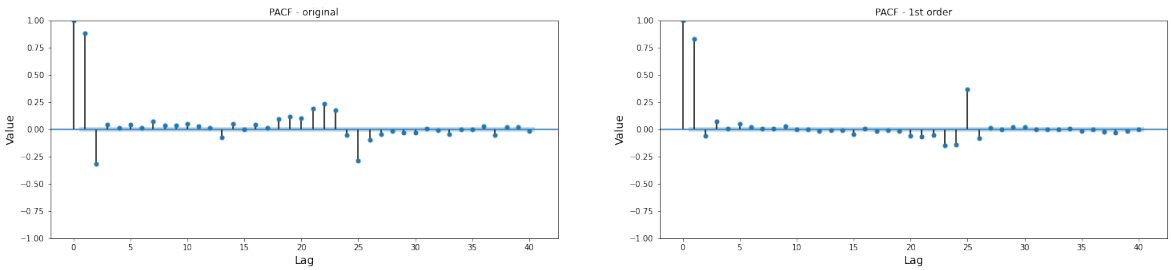

Figure 6: PACF della serie

4.3 Parametri

Per la scelta dei parametri p e q si è utilizzato un approccio automatico che ha permesso di trovare la miglior combinazione secondo il criterio della minimizzazione del AIC, tale approccio è disponibile attraverso la libreria python pmdarima che mette a disposizione il metodo `auto_arima`. Al termine di quest'analisi si sono dunque ottenuti i valori dei parametri necessari per fittare il modello che in questo caso equivalgono a `p = 2, d = 1, q = 1`.
Ottenuti i parametri ideali si è potuto fittare il modello con i dati a disposizione, in particolare sono stati realizzati diversi modelli più o meno efficienti, nei prossimi paragrafi verranno illustrati nel dettaglio quelli maggiormente significativi.

Il modello fittato seguendo gli step appena descritti risulta essere valido dal punto di vista statistico, infatti dai grafici di diagnostica, riportati in figura 7, risultano correlazioni all'interno della banda di normalità e un buon adattamento alla distribuzione normale anche se nelle code qualche osservazione tende ad allontanarsi.

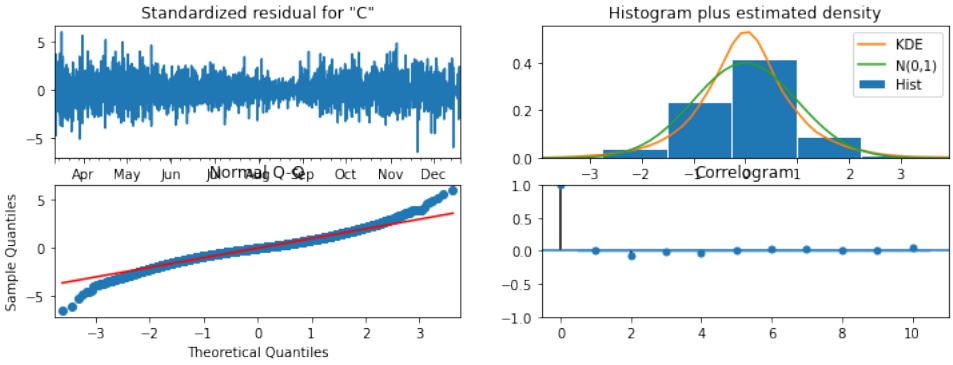

Figure 7: Diagnostica della serie storica

Per valutare la bontà del modello. Si è definita la metrica che ci permetterà di misurare quanto le predizioni del modello si avvicinano ai valori corretti, Mean Absolute Error (MAE) e la metrica MAPE(è la media degli errori percentuali). Si ottengo le seguenti performance: MAE train: 58000000, MAE validation: 149200000,MAPE train 5.167733, MAPE test 13.6444612 La predizione della serie è la seguente:

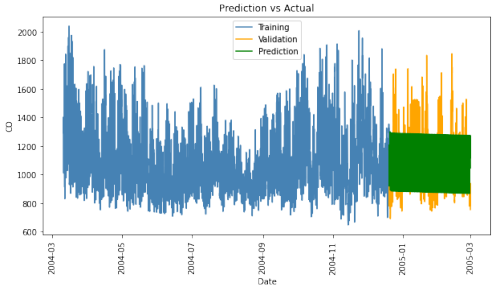

Figure 8: Previsione Arima

4.4 Analisi di fourier

Per corregere la stagionalità, sono stati aggiunti regressori esterni al modello,tramite la serie di fourier. Sono stati testati diverse componenti e combinazioni, il modello migliore risulta essere: periodo 168 (settimanale) e 2 sinusoidi. Non si hanno molti miglioramenti in termini di MAE (MAE train: 58.000000, MAE validation: 149.200000)e MAPE train 5.167731, MAPE test 13.644492.

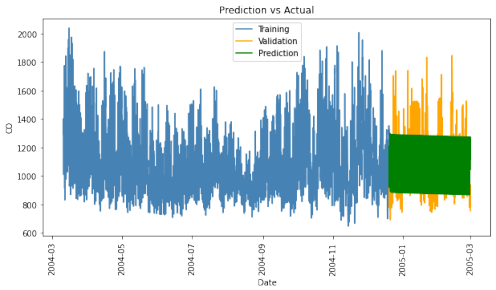

Figure 9: Previsione fourier

5 UCM

Sono stati effetuati diversi test ai modelli e confrontati con diversi parametri. Il modello migliore risulta essere composto con le seguenti componenti:

- Stagionalità Stocastica di tipo Dummy con s1=24
- Stagionalità Trigonometrica con s2=168, t con t=s1,s2
- Ciclo

La componente ciclica ha lo scopo di catturare gli effetti ciclici in intervalli di tempo molto più lunghi di quelli catturati dalla componente stagionale. I modelli confrontati sono stati : local linear trend (LLT), random walk (RW), random walk con drift e livello. Tra tutti i modelli si è scelto il modello con il valore più basso di MAE sul test, risulta essere il livello che verrà utilizzato per effettuare la previsione.

	train_mape	test_mape	train_mae	test_mae
LLTREND	5.593247	20.839465	62.782886	216.799387
RWDRIFT	5.600248	20.915399	62.869201	217.588371
LLEVAL	5.603663	19.727187	62.931187	205.362375
RW	5.616263	19.773814	63.087404	205.847707

Figure 10: UCM

6 Machine Learning

6.1 LSTM

Per quest' ultimo task si è scelto di implementare una rete neurale ricorrente di tipo LSTM per modellare al meglio la serie anche con modelli del tipo non lineare. Sono state effetuate diverse fasi di preprocessing:

- estrazione dell'array NumPy dal dataframe, convertire i valori interi in valori in virgola mobile, che sono più adatti per la modellazione con una rete neurale.
- ridimensionamento dei dati nell'intervallo da 0 a 1,ovvero normalizzazione, attraverso la preelaborazione MinMaxScaler della libreria scikit-learn.

Dopo aver modellato i dati e stimato l'abilità del nostro modello sul set di dati di addestramento, è stato necessario testare il modello su nuovi dati, è stato utilizzato perciò l'approccio cross validation. Successivamente è stato diviso il dataset in training 80 e 20Modello finale composto da due argomenti:

- il set di dati, che è un array NumPy da convertire in un set di dati.
- lookback, che è il numero di passaggi temporali precedenti da utilizzare come variabili di input per prevedere il periodo di tempo successivo, in questo caso predefinito a 168.

Questa impostazione predefinita creerà un set di dati in cui X valore di ossido di carbonio in un dato momento (t) e Y è il valore in un momento successivo (t + 1). Sono state testate diverse combinazioni: La rete presenta 256 neuroni nel primo strato e uno strato di output che effettua una previsione di valore singolo. La funzione di attivazione sigmoidea predefinita viene utilizzata per i blocchi LSTM(LSTM sono sensibili alla scala dei dati di input, in particolare quando vengono utilizzate le funzioni di attivazione sigmoid. La rete viene addestrata per 20 epoche e viene utilizzata una dimensione batch di 32. Dal grafico notiamo che il modello ha svolto un ottimo lavoro adattando sia i set di dati di addestramento che quelli di test.

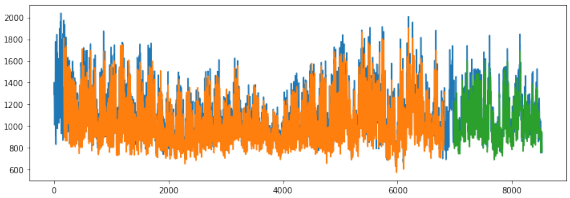

Figure 11: Previsione del modello LSTM

6.2 Gru

Successivamente al modello LSTM è stato testato un ulteriore modello ovvero GRU per determinare il migliore in termini di mae della famiglia ML. GRU impara a catturare le dipendenze dai dati passati su scale temporali differenti. Sono state effettuale le simili trasformazione citate sopra ovvero: 1 Normalizzazione del dataset: funzione MinMaxScaler con range(0,1); 2. Nuova divisione in train e test (con proporzione identica alla precedente) 3. Rimodellazione dei dataset con finestra di lookback. La rete presenta 256 neuroni nel primo strato e uno strato di output che effettua una previsione di valore singolo.
Gli iperparametri scelti per l'addestramento della rete sono:

- Loss function: categorical mean squared error
- optimizer: adam
- numero di epoche: 20
- batch size: 32

è stata utilizzata la tecnica di Earlystopping, per interrompe l'allenamento quando la metrica monitorata aveva smesso di migliorare. Dal grafico notiamo che il modello ha svolto un ottimo lavoro adattando sia i set di dati di addestramento che quelli di test.

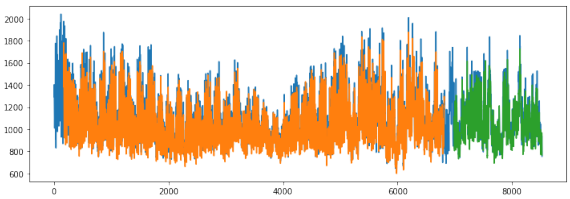

Figure 12: Previsione del modello GRU

Nella seguente figura viene mostrato le performance di tutti i modelli descritti sopra.

	train_mae	test_mae	train_mape	test_mape
ARIMA	58.000000	149.200000	5.167733	13.644612
ARIMA_FOURIER	58.000000	149.200000	5.167731	13.644492
UCLM	62.931187	205.362375	5.603663	19.727187
LSTM	58.900000	61.300000	5.188175	5.366033
GRU	55.000000	54.900000	4.868371	4.874036

Figure 13: Performance dei modelli

7 Previsioni orarie dal 2005-03-01 al 2005-03-31

Le predizioni della mese richiesto sono le seguenti:

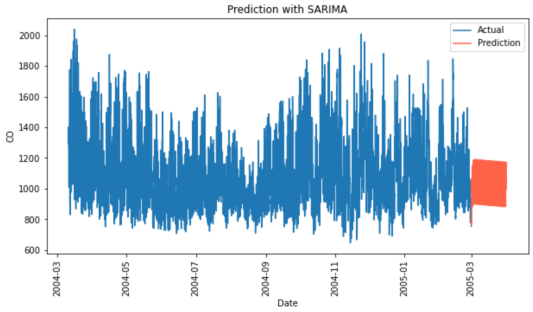

Figure 14: Prediction sarimax

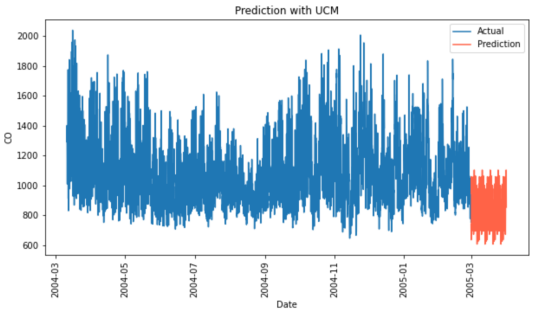

Figure 15: Prediction UCLM

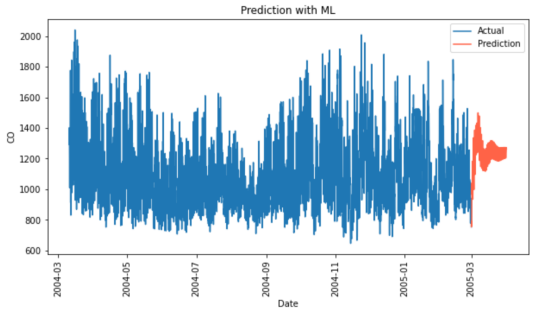

Figure 16: Prediction ML

	Data	ARIMA	UCM	ML
0	2005-03-01 00:00:00	755.000000	740.099328	793.670432
1	2005-03-01 01:00:00	790.237966	686.591961	785.894946
2	2005-03-01 02:00:00	783.811428	643.911856	770.008173
3	2005-03-01 03:00:00	768.492611	636.716183	767.212427
4	2005-03-01 04:00:00	768.006609	644.285111	769.863930
...	...	...	...	...
739	2005-03-31 19:00:00	1263.637757	1103.504733	1152.989555
740	2005-03-31 20:00:00	1270.423905	1101.413261	1172.641242
741	2005-03-31 21:00:00	1272.280357	973.671987	1098.370507
742	2005-03-31 22:00:00	1269.178425	896.435753	1046.435206
743	2005-03-31 23:00:00	1263.838189	855.179011	1039.730270

744 rows × 4 columns

Figure 17: Prediction final

8 Conclusioni

In conclusione il modello migliore è il modello di machine learning(GRU),le previsioni sono abbastanza buone, si dovrebbero avere maggior dati e anni per un' analisi più approfondita.