POLITECNICO DI TORINO

Corso di Laurea
in Ingegneria Matematica

Approcci statistici per la modellizzazione e previsione della concentrazione di ossigeno disciolto nella SmartBay di Santa Teresa

Relatori
Gianfranco Durin
prof. Gianluca Mastrantonio
prof. Francesco Vaccarino

Candidata
Matilde Pattarino

Anno Accademico 2024-2025

Indice

Sommario

Gli oceani e i loro ecosistemi sono tra gli habitat naturali che risentono maggiormente dei cambiamenti climatici. Un indicatore determinante del loro stato di salute è l'ossigeno disciolto in acqua, fondamentale per il metabolismo aerobico degli organismi marini.
Risulta quindi determinante misurare e prevedere la sua concentrazione nell'acqua marina.
La presente tesi si inserisce nel contesto del progetto "SmartTwin: Oxygen Digital Twin di Smart Bay S. Teresa", finanziato dalla Fondazione CRT di Torino, a cui partecipano l'Istituto Nazionale di Ricerca Metrologica (INRiM) e il Centro Ricerche Ambiente Marino S. Teresa (ENEA) di Lerici in provincia di La Spezia.
L'obiettivo del progetto è quello di realizzare un gemello digitale della baia, in grado di replicare e prevedere le dinamiche e le variabili ambientali locali dell'ecosistema. Questo è possibile grazie all'analisi dei dati provenienti da alcuni sensori disposti nella baia. Le misurazioni, a cadenza oraria, comprendono diversi parametri oltre alla concentrazione di ossigeno disciolto, tra cui temperatura dell'acqua, pressione e salinità. Inoltre, la stazione metereologica di ENEA fornisce dati atmosferici quali la temperatura dell'aria, l'intensità della radiazione solare e le precipitazioni.
Per l'elaborazione e la manipolazione dei dati raccolti, il focus principale di questa tesi sarà sull'analisi delle serie temporali, e a tal proposito sarà fornito un background matematico. Successivamente, si procederà con la spiegazione dei metodi statistici utilizzati per la modellizzazione e la previsione dei dati: il modello SARIMA (Seasonal AutoRegressive Integrated Moving Average), e il modello Prophet.
Il modello SARIMA è un'estensione del modello ARIMA, progettata per gestire andamenti periodici o stagionali. Il modello Prophet, sviluppato da Meta, adotta invece un approccio basato sulla scomposizione della serie temporale nelle sue principali componenti: il trend, ovvero la tendenza a lungo termine, la stagionalità, il rumore bianco e le cosiddette "holidays", degli eventi anomali che non seguono un pattern preciso ma che possono influenzare i valori della serie. In particolare, nella dinamica locale della baia, sono state considerate come holidays i giorni di piena dei fiumi Arno e Magra, che sfociano nelle vicinanze della baia, e i giorni in cui la radiazione solare è particolarmente bassa. Questo perché l'ossigeno disciolto non è solo il risultato dell'interazione delle variabili fisiche, ma dipende fortemente dal ciclo di fotosintesi da parte del fitoplancton, presente in baia.
Per valutare le performance di tali approcci, sia nel caso univariato che nel caso multivariato, ovvero con l'aggiunta di regressori, sono state valutate le previsioni ottenute su un intervallo temporale di 3 giorni considerando diverse metriche di errore e i tempi di addestramento dei modelli.
I risultati mostrano che le prestazioni dei modelli SARIMA e Prophet sono comparabili in termini di accuratezza predittiva, con un miglioramento significativo quando vengono inclusi i regressori nel modello Prophet. Tuttavia, Prophet si distingue per una maggiore efficienza computazionale, rendendolo particolarmente adatto a scenari in cui è richiesto un aggiornamento frequente delle previsioni.

Capitolo 1

Introduzione

Negli ultimi decenni, la consapevolezza dell'impatto che hanno le attività umane sull'ambiente è cresciuta in modo drastico.
La comunità scientifica concorda sul fatto che il nostro pianeta si stia scaldando così velocemente a causa dei gas serra (principalmente anidride carbonica, metano e ossido di azoto) intrappolati nell'atmosfera, e che questi gas siano prodotti dalle attività umane quali uso di combustibili fossili (per energia, industrie, trasporti), deforestazione, utilizzo di fertilizzanti azotati in agricoltura e pratiche di allevamento intensivo.
La maggior parte delle persone ha potuto vedere coi propri occhi gli effetti del cambiamento climatico, ma non si tratta solo di eventi estremi come alluvioni, siccità, inondazioni, bensì esistono altre variazioni minori e più lente che potrebbero cambiarci la vita nel giro di pochi anni o decenni.
La buona notizia è che sia la lotta ai cambiamenti climatici che l'adattamento a un mondo che si riscalda sono priorità assolute per l'Unione Europea. Ne è stato un esempio il Green Deal europeo, varato nel 2019 dalla presidente Ursola von der Leyen, che ha come obiettivo generale quello di raggiungere la neutralità climatica in Europa entro il 2050, con obiettivi intermedi come la riduzione delle emissioni del 55% entro il 2030.

1.1 Il ruolo dell'ossigeno disciolto nell'oceano

Particolare attenzione va rivolta agli effetti dei cambiamenti climatici sugli oceani, i quali assorbono una quota significativa del carbonio antropico emesso nell'atmosfera e gran parte del calore in eccesso dovuto all'effetto serra. Tali processi hanno portato a due problemi strettamente correlati: da un lato il surriscaldamento delle acque oceaniche, dall'altro l'innalzamento del livello del mare, dovuto allo scioglimento dei ghiacciai continentali, della Groenlandia e di parte dell'Antartide. Questi fenomeni comportano gravi ripercussioni sugli ecosistemi marini e costieri, contribuendo alla perdita di biodiversità, all'acidificazione e alla deossigenazione degli oceani. Di conseguenza le capacità benefiche e preziose che l'oceano offre sono sempre più a rischio.

Infatti l'ossigeno disciolto [3], ovvero l'ossigeno allo stato molecolare (O_2) presente in forma libera nell'acqua, rappresenta una delle variabili più rilevanti per la valutazione dello stato ecologico degli oceani e degli ecosistemi acquatici. La sua presenza è infatti essenziale per il metabolismo aerobico di pesci, crostacei, plancton e microrganismi, che dipendono direttamente da questo gas per produrre energia tramite la respirazione cellulare.

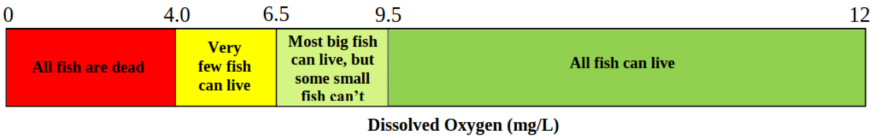


Figura 1.1: Livello di ossigeno disciolto in mg/l necessario per la sopravvivenza dei pesci.

1.1.1 Parametri che influenzano la concentrazione di ossigeno

La concentrazione di ossigeno disciolto in acqua rappresenta l'equilibrio tra la sua produzione, da parte di fitoplancton e alghe tramite la fotosintesi, e il suo consumo, dovuto alla respirazione degli organismi, ma non solo. Infatti la presenza di ossigeno è anche il risultato dell'influenza di diversi parametri ambientali:

- Temperatura: ha un ruolo fondamentale, poiché la solubilità dei gas, in generale, diminuisce all'aumentare della temperatura.
- Salinità: altrettanto importante, per il motivo che acque più salate contengono meno ossigeno rispetto a quelle dolci. Tale fenomeno, detto "effetto salting-out", è causato dalla competizione tra le molecole di gas (O_2) e gli ioni di sale disciolti (Na^+ e Cl^-) per le interazioni con le molecole d'acqua
- Pressione atmosferica: siccome influenza la pressione parziale dell'ossigeno, maggiore è la pressione, maggiore sarà la quantità di ossigeno che si scioglie nell'acqua.

Nello specifico, la legge di Henry modellizza la concentrazione dell'ossigeno C_{O_2} in acqua in questo modo:

$$C_{O_2} = k_H(T_{H_2O}, S_{H_2O}) \cdot P_{O_2},$$

dove:

- $k_H(T_{H_2O}, S_{H_2O})$ è la costante di Henry, che dipende dalla temperatura e dalla salinità dell'acqua;
- P_{O_2} è la pressione parziale dell'ossigeno in aria, che a sua volta è data da $P_{O_2} = x_{O_2} \cdot P_{\text{tot}}$, ovvero dal prodotto tra la la frazione di ossigeno nell'aria e la pressione totale che l'aria esercita sulla superficie dell'acqua

Il passaggio naturale tra la teoria e la pratica è rappresentato dallo studio di Weiss [17], che ha condotto uno studio empirico basato su misure sperimentali della solubilità dell'ossigeno a diverse temperature e salinità dell'acqua. La formula ottenuta esprime il logaritmo

naturale della concentrazione di ossigeno puro in acqua (misurato in ml/kg) a pressione P_{O_2} di 1 atm, in funzione di temperatura (in Kelvin) e salinità (in PSU):

$$\ln(C^*_{O_2}) = A_1 + A_2\left(\frac{100}{T}\right) + A_3\ln\left(\frac{T}{100}\right) + A_4\left(\frac{T}{100}\right)^2 + S\left(B_1 + B_2\frac{T}{100} + B_3\left(\frac{T}{100}\right)^2\right)$$

con i coefficienti A_i e B_i stimati in [12] In definitiva, la formulazione che restituisce la concentrazione dell'ossigeno, è data da:

$$C_{O_2} = P_{O_2}\exp\left[A_1 + A_2\left(\frac{100}{T}\right) + A_3\ln\left(\frac{T}{100}\right) + A_4\left(\frac{T}{100}\right)^2 + S\left(B_1 + B_2\frac{T}{100} + B_3\left(\frac{T}{100}\right)^2\right)\right]$$

I valori che si ottengono si riferiscono al cosiddetto *ossigeno termodinamico*, il valore teorico di equilibrio che dipende solo dalle variabili fisiche dell'ambiente, a differenza della concentrazione reale, che riflette l'interazione complessa tra termodinamica, idrodinamica e biologia. Tale valore rappresenta quindi la quantità massima di ossigeno che l'acqua può contenere date certe condizioni di temperatura, salinità e pressione, e costituisce il riferimento per il calcolo della saturazione dell'ossigeno.
Quest'ultima indica quanto la concentrazione osservata si avvicina o si discosta dal valore di equilibrio termodinamico e si esprime come rapporto tra l'ossigeno misurato e quello di saturazione:

$$\text{Saturazione (\%)} = \frac{C_{O_2\,\text{misurato}}}{C_{O_2\,\text{saturazione}}} \cdot 100$$

In condizioni standard, la saturazione varia tra l'85 % e il 105 %. Tuttavia, in presenza di intensa attività fotosintetica si possono registrare fenomeni di sovrasaturazione, in cui l'ossigeno disciolto supera il valore teorico massimo [3]. Al contrario, quando il consumo biologico e chimico prevale sulla produzione o sullo scambio con l'atmosfera, l'acqua può trovarsi in condizioni di sottosaturazione.

1.2 Il progetto SmartTwin

Una delle iniziative più recenti a livello internazionale in materia di tutela degli ecosistemi marini, è il One Ocean Summit [13]: un meeting internazionale tenuto nel 2022 a Brest, con l'obiettivo di sollecitare l'ambizione e l'impegno della comunità internazionale in materia di protezione marina e tradurre la responsabilità condivisa verso l'oceano in impegni concreti.
Tra i numerosi accordi raggiunti, alcuni riguardano direttamente la gestione integrata e la conoscenza del mare: in particolare l'Unione Europea si è impegnata nell'ambizioso progetto di realizzare un "Gemello Digitale dell'Oceano" (European Digital Twin of the Ocean, EDTO), uno strumento innovativo che mira a creare un modello virtuale dinamico dell'oceano per migliorare la conoscenza marina, supportare decisioni politiche basate su dati scientifici e promuovere un'economia blu sostenibile.
Il programma è coordinato e gestito dalla Commissione Europea in collaborazione con diversi enti di ricerca e organizzazioni internazionali. Esso si basa sull'elaborazione di

grandi volumi di dati provenienti da satelliti, da reti di osservazione terrestri, aeree e marine, nonché da modelli numerici e simulativi. Tali dati vengono continuamente integrati e aggiornati, consentendo una rappresentazione digitale coerente e in tempo quasi reale dello stato e dell'evoluzione dell'oceano.

Parallelamente a questa iniziativa europea, si colloca il progetto "SmartTwin: Oxygen Digital Twin di Smart Bay S. Teresa", oggetto di questa tesi, sviluppato dall'INRiM (Istituto Nazionale di Ricerca Metrologica), che collabora con ENEA (Agenzia nazionale per le nuove tecnologie, l'energia e lo sviluppo economico sostenibile) di Lerici, in provincia di La Spezia in Liguria.
Insieme, queste realtà ambiscono alla realizzazione di un gemello digitale della baia (Smart Bay [15]) di Santa Teresa di Lerici, grazie a dei sistemi di monitoraggio e comunicazioni sottomarine. Lo scopo è quello di creare una replica virtuale dinamica dell'ambiente marino locale, in grado di riprodurre e prevedere quasi in tempo reale le principali variabili che caratterizzano la baia, in particolar modo la concentrazione di ossigeno disciolto. Tale strumento permetterebbe di comprendere in modo approfondito i processi che avvengono nella baia e di valutare lo stato di salute del mare, al fine di identificare tepestivamente eventuali condizioni anomale e di potenziale rischio.

L'obiettivo di questa tesi è quello di valutare e confrontare le performance di due modelli statistici applicati ai dati relativi all'ossigeno disciolto nella baia, nell'ottica sviluppare un metodo per fare previsioni future su 3 giorni.
In particolare, verranno valutati due modelli differenti:

- il modello SARIMA (Seasonal AutoRegressive Integrated Moving Average), di natura classica e statistica, basato sull'analisi delle componenti autoregressive, di media mobile e stagionali della serie temporale;

- il modello Prophet, un approccio più flessibile sviluppato da Meta, che sfrutta una decomposizione additiva della serie nelle sue componenti di trend, stagionalità e *holidays*, degli eventi anomali che non seguono un pattern preciso ma che possono influenzare i valori della serie.

Entrambi i modelli, oltre alla formulazione univariata tradizionale, consentono l'estensione all'approccio multivariato, in cui la variabile di interesse (in questo caso, l'ossigeno disciolto) viene prevista tenendo conto di grandezze esplicative aggiuntive che influenzano il suo comportamento, come temperatura dell'acqua, salinità, e radiazione solare.
La valutazione sarà condotta attraverso diverse metriche di accuratezza e confrontando i tempi di addestramento e di previsione dei modelli, al fine di evidenziare non solo la precisione ma anche l'efficienza computazionale delle due soluzioni, parametro fondamentale in prospettiva di un utilizzo operativo all'interno del gemello digitale.

1.3 I dati

1.3.1 Raccolta dei dati

I sensori utilizzati per le rilevazioni sono disposti nella baia in 4 punti ("Baia" in rosso, "Mitili" in nero, "Boas" in verde, "Tinetto" in blu), come in figura:

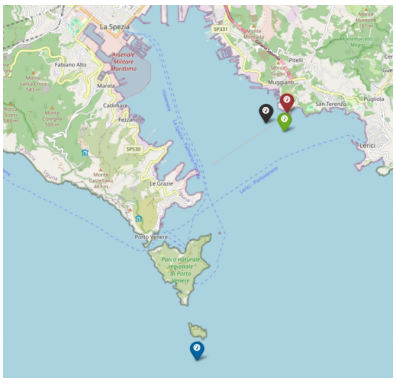


Figura 1.2: Posizioni dei sensori.

Ogni sensore rileva i seguenti parametri ogni 30 minuti:

- concentrazione di ossigeno disciolto in acqua ($\mu mol/l$);
- pressione ($dBar$), per monitorare variazioni nella colonna d'acqua dovute a maree e condizioni atmosferiche;
- temperatura ($°C$), parametro fondamentale per lo studio della solubilità dell'ossigeno;
- salinità (PSU), che, insieme alla temperatura, permette di calcolare la densità dell'acqua e la conseguente concentrazione di ossigeno;
- conducibilità (mS/cm), che dipende dalla concentrazione di ioni e dalla temperatura dell'acqua. Si tratta di un indicatore diretto della salinità;
- profondità (m) a cui si trova il sensore, sempre attorno a 1 metro;
- voltaggio (V) della batteria dei sensori.

La veridicità di questi dati può talvolta essere influenzata dagli eventi di biofouling, ovvero l'accumulo indesiderato di organismi, piante e microrganismi sui sensori e dalla conseguente pulizia degli stessi, oltre che essere affetta dal livello di batteria nel momento della rilevazione.
Per poter calibrare al meglio i dati e riempire eventuali buchi, con cadenza settimanale vengono calate in mare delle sonde di tipo CTD probe, che contengono sensori molto affidabili che misurano conducibilità (C), temperatura (T) e profondità (D).

In particolare, la serie relativa all'ossigeno disciolto utilizzata in questa trattazione non deriva direttamente dalle misure grezze fornite dai sensori, ma da una versione "riempita" mediante un modello multi-stage di ricostruzione dei dati, descritto in dettaglio in [2]. Tale approccio consente di correggere eventuali discontinuità dovute a problemi strumentali e di ottenere una serie temporale più coerente e completa, adatta alle analisi successive.

A questi dati si aggiungono informazioni sulle condizioni atmosferiche provenienti dalla stazione metereologica di ENEA, che si trova sulla punta Santa Teresa a 50 metri sopra al livello del mare. In particolare, la stazione rileva parametri quali:

- temperatura dell'aria ($°C$);
- radiazione solare (W/m^2);
- velocità del vento (m/s);
- pressione atmosferica (hPa);
- umidità relativa (%);
- precipitazioni (mm).

1.3.2 Caratteristiche dei dati

Questa trattazione utilizzerà i dati "riempiti" che provengono dal nodo "Baia". Tali misurazioni sono campionate in modo da avere cadenza oraria e riguardano il periodo temporale che va dal 25 marzo 2021 al 22 novembre 2023, per un totale di più di due anni e mezzo. Le rilevazioni sono state raccolte nel dataframe Pandas, riportato di seguito:

Time	T_Water(°C)	Salinity(PSU)	Depth(m)	Conductivity(mS/cm)	Pressure(db)	Voltage(V)	Wind_speed(m/s)	T_Air(°C)	Solar_Radiation(W/m^2)	Atm_Pressure(hPa)	Humidity(%)	Rain(mm)	Oxygen_wsense(umol/l)	Oxygen_wsense(%)
2021-03-25 11:00:00	13.997104	37.320825	NaN	44.409741	1.209811	NaN	2.845417	10.033553	4040.133333	1029.363056	76.369722	0.000000	256.565174	100.495686
2021-03-25 12:00:00	14.050000	37.247719	NaN	44.386696	1.209330	NaN	2.921354	10.648220	4517.733333	1029.363056	76.160069	0.000000	258.178013	101.182313
2021-03-25 13:00:00	14.290000	36.935375	NaN	44.300464	1.205035	NaN	3.078958	11.280720	4452.333333	1029.342222	76.171667	0.000000	257.691716	101.253515
2021-03-25 14:00:00	14.385000	36.801858	NaN	44.254734	1.205082	NaN	3.134271	11.906053	3975.400000	1029.342222	76.349375	0.000000	258.323074	101.598346
2021-03-25 15:00:00	14.385000	36.825386	NaN	44.279969	1.206289	NaN	3.189896	12.512387	3094.333333	1029.342222	76.643681	0.000000	258.881051	101.834416
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2023-11-22 04:00:00	16.973574	36.915203	NaN	47.054901	0.998009	NaN	4.840521	13.763220	0.000000	1012.182500	62.758403	0.036111	219.828702	90.964288
2023-11-22 05:00:00	16.971628	36.924103	NaN	47.062969	0.996870	NaN	4.921979	13.789887	0.000000	1012.231111	61.020486	0.036111	219.728560	90.925015
2023-11-22 06:00:00	16.946977	36.947290	NaN	47.063535	1.007262	NaN	4.932812	13.817220	5.420667	1012.335278	59.033403	0.036111	222.532795	92.057065
2023-11-22 07:00:00	16.936029	36.958549	NaN	47.064868	1.014752	NaN	4.814896	13.835720	91.810667	1012.460278	57.098750	0.036111	225.359451	93.214244
2023-11-22 08:00:00	16.940400	36.982814	NaN	47.096947	1.017914	NaN	4.721354	13.875553	292.977333	1012.626944	55.164306	0.036111	228.335891	94.468778

23326 rows × 14 columns

Figura 1.3: Dataframe Pandas coi dati provenienti dal sensore dispopsto nel nodo "Baia" e dalla stazione metereologica di ENEA.

L'ultima colonna, "Oxygen_wsense(%)", si riferisce all'ossigeno di saturazione, trattato nella sezione 1.1.1 e calcolato rispetto al dato misurato della penultima colonna, "Oxygen_wsense($\mu mol/l$)".
Di seguito è riportato il grafico di alcuni dati significativi disponibili e, successivamente, lo stesso grafico limitato ai primi due mesi di osservazione, per una migliore leggibilità.

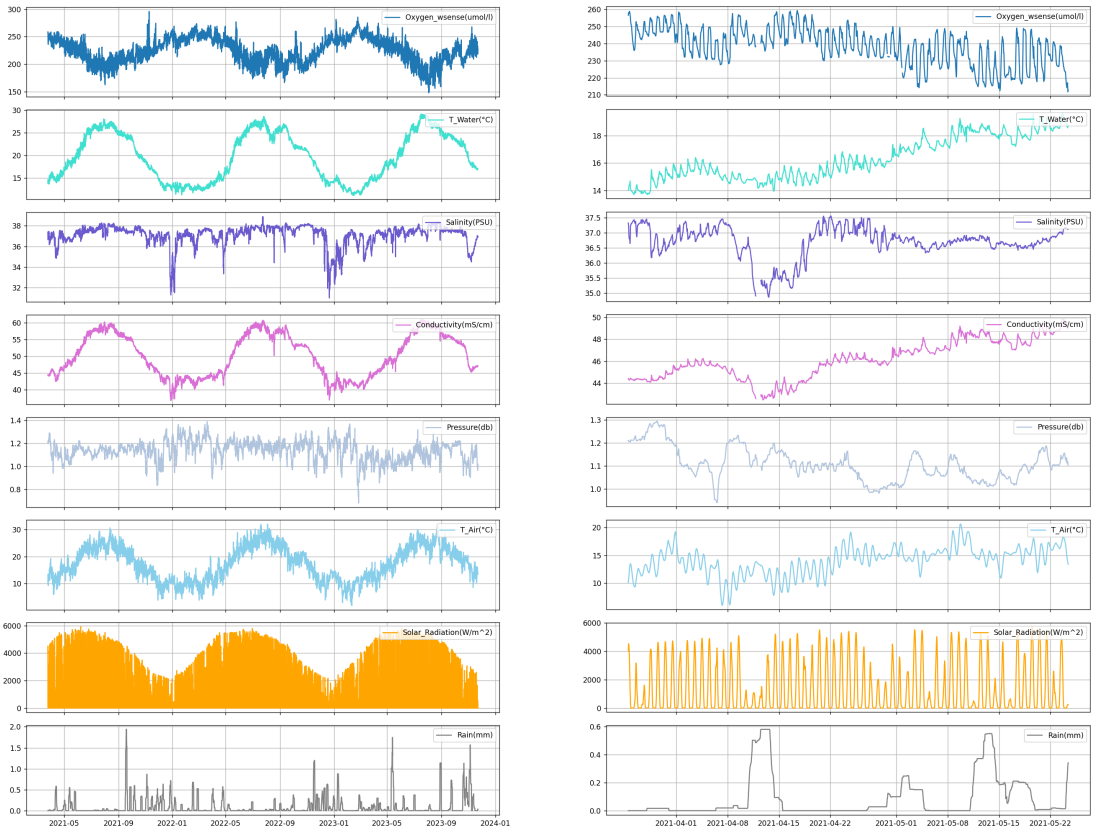


Figura 1.4: Plot dei dati su tutto il periodo e limitato ai primi due mesi di osservazione.

L'andamento temporale delle variabili evidenzia in primo luogo la presenza di due tipi di stagionalità: una annuale, maggiormente visibile nella figura sulla sinistra, dovuta ai cicli climatici e termici, e una giornaliera, chiaramente osservabile nella figura sulla destra, legata all'intensità delle radiaizoni solari e ai processi biologici di fotosintesi.
La concentrazione di ossigeno disciolto mostra valori massimi nei mesi invernali, quando le temperature sono più fredde e la solubilità viene favorita, mentre raggiunge valori minimi durante l'estate, in corrispondenza di temperature più calde. Tale comportamento riflette la legge di Henry citata in 1.1.1, secondo cui la concentrazione dei gas, come l'ossigeno in questo caso, decresce all'aumentare della temperatura. Quest'ultima infatti, mostra una stagionalità inversa rispetto a quella dell'ossigeno, sia nel caso della serie inerente alla temperatura dell'acqua sia che relativa all'aria.
La conducibilità ha un andamento molto simile a quello delle temperature, con valori elevati nei mesi estivi e valori più moderati in quelli invernali. È strettamente legata alla salinità, anche se è meno chiaro visivamente. Le oscillazioni della salinità sono più contenute e presentano dei cali bruschi in corrispondenza di eventi di precipitazioni intense, visibili nell'ultimo subplot.
Infine, la pressione presenta una variabilità irregolare, priva di una stagionalità evidente, che però comunque aggiunge il suo contributo all'effettivo valore di ossidegno disciolto.

Correlazione

Tali dipendenze sono state confermate dall'analisi delle correlazioni rispetto alla concentrazione di ossigeno disciolto, tralasciando l'ossigeno di saturazione. Per arricchire l'analisi, sono state introdotte due nuove variabili, maggiormente significative rispetto a quelle di partenza, ottenute tramite rolling: la prima rappresenta la pioggia cumulata nelle ultime 72 ore [2] e la seconda esprime la radiazione solare cumulata nelle due ore precedenti.

Variabile	**Coefficiente di correlazione**
Atm_Pressure(hPa)	0.241
Solar_Radiation_sum_2h	0.093
Solar_Radiation(W/m^2)	0.084
Depth(m)	0.044
rain_sum_72h	0.041
Pressure(db)	0.040
Wind_speed(m/s)	0.004
Rain(mm)	0.001
Humidity(%)	-0.037
Voltage(V)	-0.129
Salinity(PSU)	-0.497
T_Air(°C)	-0.683
T_Water(°C)	-0.836
Conductivity(mS/cm)	-0.837

Tabella 1.1: Correlazioni tra la concentrazione di ossigeno disciolto e le altre variabili.

Si riporta anche l'heatmap corrispondente:

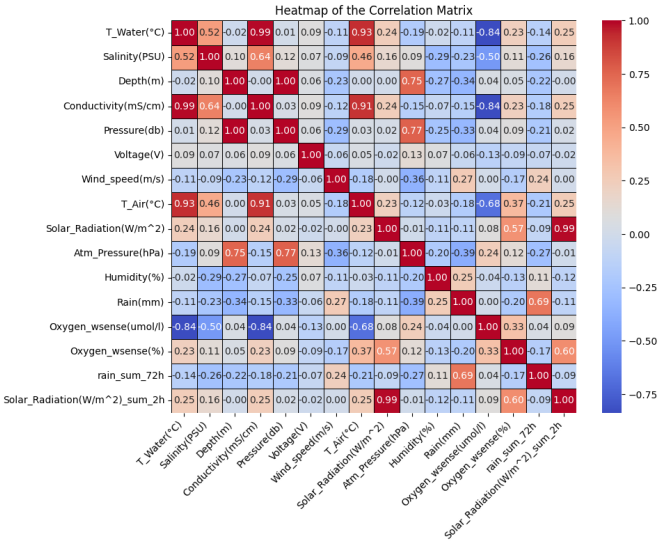


Figura 1.5: Heatmap della matrice di correlazione tra le variabili

Le variabili più influenti, correlate positivamente o negativamente, risultano essere le temperature, la conducività, la salinità e la pressione atmosferica.

Stagionalità

Nel contesto della stagionalità giornaliera, gioca un ruolo fondamentale la radiazione solare, che presenta un'andamento sinuisoidale con ampiezza massima nei mesi estivi, ma anche oscillazioni giornaliere dovute al ciclo solare. Durante le ore diurne, la concentrazione di ossigeno tende ad aumentare grazie alle attività fotosintetiche del fitoplancton, mentre durante la notte prevale la respirazione e il consumo di ossigeno.
Si prende come esempio il confronto tra il mese di gennaio 2022 e il mese di giungno dello stesso anno:

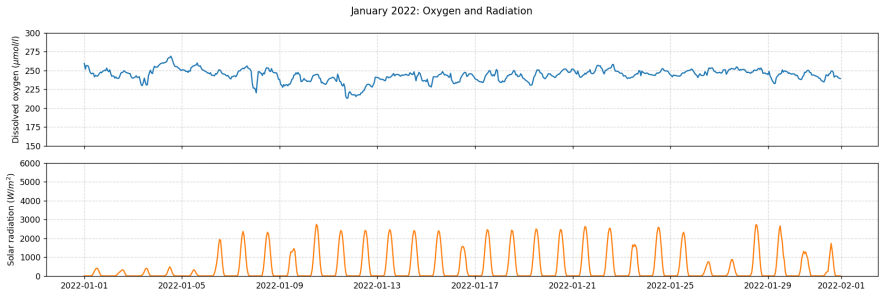


Figura 1.6: Ossigeno disciolto e radiazioni solari nel mese di gennaio 2022.

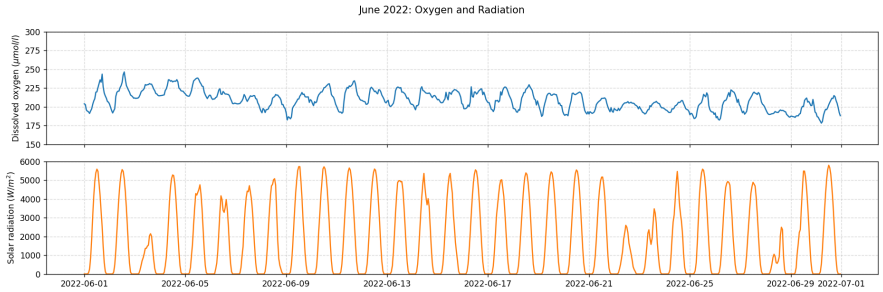


Figura 1.7: Ossigeno disciolto e radiazioni solari nel mese di giugno 2022.

È evidente come, nel mese di giugno, l'andamento dell'ossigeno disciolto oscilli molto di più su scala giornaliera, seguendo l'alternarsi del giorno e della notte.

Il comportamento ciclico giornaliero osservato nella concentrazione di ossigeno disciolto in periodi diversi dell'anno trova un riscontro quando si confronta l'ossigeno reale osservato e l'ossigeno di saturazione, calcolato sulla base delle variabili fisiche marine e trattato nella sezione 1.1.1.
Si riportano di seguito le due serie temporali: in blu quella relativa alla concentrazione di ossigeno misurato e in nero quella relativa all'ossigeno di saturazione. In quest'ultima, il valore 100 indica la condizione di equilibrio termodinamico, valori superiori a 100% corrispondono a sovrasaturazione mentre valori inferiori a 100% indicano sottosaturazione.

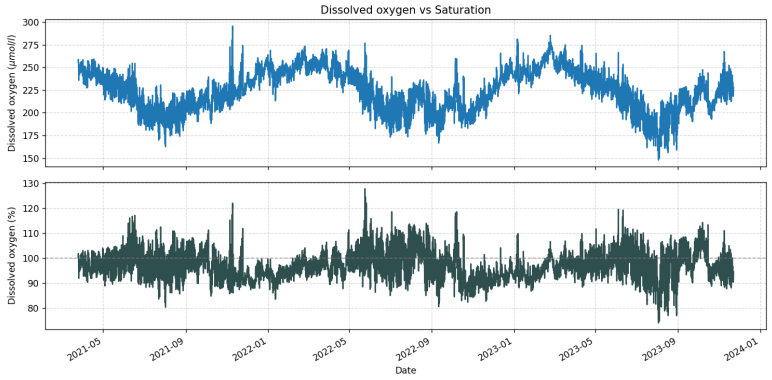


Figura 1.8: Confronto tra concentrazione e saturazione dell'ossigeno disciolto.

La discrepanza tra i due valori permette di comprendere quanto l'ambiente marino si discosti dalle condizioni di equilibrio e di evidenziare l'influenza dei processi locali biologici, in particolare dell'attività del fitoplancton. Tali dinamiche ora risultano evidenti anche su scala annuale: nei mesi estivi, l'aumento della radiazione solare favorisce la crescita del fitoplancton e, di conseguenza, periodi di sovrasaturazione più prolungata e marcata. Viceversa, durante l'inverno, la riduzione della radiazione e la minore attività fotosintetica determinano valori medi di ossigeno reale più prossimi o inferiori a quelli termodinamici.

Per poter apprezzare contemporaneamente la stagionalità giornaliera e annuale dell'ossigeno disciolto si riporta anche il Seasonal Plot, in figura 1.9.

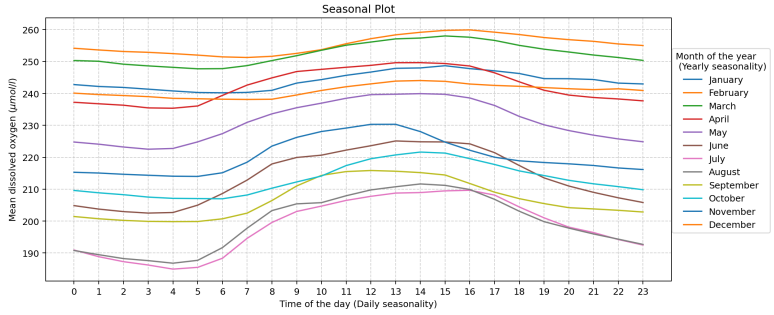


Figura 1.9: Seasonal Plot dell'ossigeno disciolto.

Nel grafico si osserva un andamento sinusoidale nell'arco delle 24 ore, con valori minimi nelle prime ore del mattino e massimi nelle ore pomeridiane, riflettendo l'alternanza tra respirazione notturna e fotosintesi diurna del fitoplancton.
Oltre alla ciclicità giornaliera, si nota anche l'alternanza annuale: i mesi estivi (giugno, luglio, agosto, settembre) presentano livelli medi più bassi di ossigeno disciolto, mentre i mesi invernali (dicembre, gennaio, febbraio, marzo) mostrano concentrazioni più elevate. Tale grafico conferma quindi la doppia periodicità del sistema.

Capitolo 2

Background matematico

2.1 Serie Temporali

Le serie temporali sono sequenze di osservazioni prese nel tempo, ad intervalli discreti ed equispaziati, nelle quali si suppone ci sia una dipendenza temporale, ovvero che le osservazioni future dipendano in qualche modo da quelle passate.
Denotando con X la variabile di interesse, ovvero la variabile di stato, si può indicare con X_t l'osservazione avvenuta al tempo t. Una serie temporale di lunghezza T può essere così espressa: $X_t = \{X_{t_1}, X_{t_2}, \ldots, X_T\}$.

Le serie temporali possono essere decomposte in quattro componenti principali:

- Trend: rappresenta l'andamento della serie nel lungo periodo, che può avere una tendenza crescente, decrescente o stazionaria;
- Stagionalità: rappresenta la componente periodica della serie, che si ripete a intervalli regolari, ad esempio con cadenza giornaliera, settimanale o annuale;
- Ciclicità: rappresenta le oscillazioni a medio-lungo termine che possono essere caratterizzate non necessariamente da un periodo fisso. Si differenzia dalla stagionalità per il fatto di non essere strettamente periodica, ma piuttosto collegata a meccanismi economici, biologici o ambientali più lenti.
- Rumore: rappresenta la componente casuale della serie, che non può essere spiegata dalle componenti precedenti.

2.1.1 Stazionarietà

La stazionarietà è una proprietà fondamentale delle serie temporali, anche perché tale caratteristica è richiesta per poter applicare molti risultati teorici e tecniche di modellazione, come i modelli ARIMA.
In sostanza, una serie è stazionaria se le sue proprietà statistiche non cambiano nel tempo: la media, la varianza e la covarianza restano costanti, e quindi le caratteristiche della serie

non dipendono dal momento in cui essa viene osservata [6].
Esistono due gradi di stazionarietà: quella forte, spesso difficile da verificare, e quella debole:

- Un processo si dice stazionario in senso forte se la distribuzione congiunta di qualsiasi insieme di osservazioni rimane invariata nel tempo, anche dopo una traslazione temporale. Ovvero, dato $\mathcal{T}$ l'insieme degli istanti temporali equispaziati e l il lag:

  $$\forall n \in \mathbb{N},\ \forall \text{ tupla } (t_1, t_2, \ldots, t_n) \in \mathcal{T}^n,\ \forall l \in \mathbb{N} : t_i + l \in \mathcal{T},\ \forall i \in \{1,2,\ldots,n\}$$
  $$\mathbb{P}(X_{t_1} = x_1, X_{t_2} = x_2, \ldots, X_{t_n} = x_n) = \mathbb{P}(X_{t_1+l} = x_1, X_{t_2+l} = x_2, \ldots, X_{t_n+l} = x_n)$$

- Un processo si dice stazionario in senso debole di ordine $s \in \mathbb{N}$ se tutti i momenti fino all'ordine s sono invarianti rispetto al tempo, ovvero non dipendono dall'istante di osservazione ma solo dal lag temporale. Formalmente, dato $\mathcal{T}$ l'insieme degli istanti temporali equispaziati e l il lag:

  $$\forall n \leq s \in \mathbb{N},\ \forall \text{ tupla } (t_1, t_2, \ldots, t_n) \in \mathcal{T}^n,\ \forall l \in \mathbb{N} : t_i + l \in \mathcal{T},\ \forall i \in \{1,2,\ldots,n\},$$
  $$\mathbb{E}[X_{t_1} X_{t_2} \ldots X_{t_n}] = \mathbb{E}[X_{t_1+l} X_{t_2+l} \ldots X_{t_n+l}]$$

  Ad esempio, per un processo stazionario in senso debole di ordine 2 valgono le seguenti condizioni:

  $$\begin{cases} \mathbb{E}[X_{t_i}] = \mathbb{E}[X_{t_i+l}] = \mu \\ \mathrm{Cov}(X_{t_i}, X_{t_i+l}) = \mathbb{E}[X_{t_i}\, X_{t_i+l}] - \mathbb{E}[X_{t_i}]\,\mathbb{E}[X_{t_i+l}] = \gamma(l) \end{cases}$$

  ossia la media è costante e la covarianza tra elementi ritardati della serie dipende unicamente dal lag temporale l tra le osservazioni. In particolare, $\gamma(\cdot)$ rappresenta la funzione di autocovarianza, che verrà approfondita nella sezione successiva 2.1.2.

Si noti che la stazionarietà in senso forte implica quella in senso debole: se infatti la distribuzione congiunta del processo rimane invariata nel tempo, ne consegue che anche la media, la varianza e la covarianza, che ne rappresentano momenti di ordine inferiore, saranno altrettanto costanti.

2.1.2 Autocovarianza e autocorrelazione

Si introducono alcune definizioni e concetti utili al fine di descrivere le proprietà statistiche delle serie temporali, che permettono di valutare il grado di dipendenza tra le osservazioni della serie [9].

Funzione di Autocovarianza

L'autocovarianza è una misura statistica che permette di calcolare la covarianza tra i valori che assume un processo in istanti diversi. In particolare, l'autocovarianza di ordine l della variabile X_{t_i} è definita come:

$$\begin{aligned} \gamma_{t_i}(l) = \mathrm{Cov}(X_{t_i}, X_{t_i+l}) &= \mathbb{E}\left[(X_{t_i} - \mathbb{E}[X_{t_i}])(X_{t_i+l} - \mathbb{E}[X_{t_i+l}])\right] \\ &= \mathbb{E}[X_{t_i} X_{t_i+l}] - \mathbb{E}[X_{t_i}]\mathbb{E}[X_{t_i+l}] \end{aligned}$$

dove l è il lag, il ritardo con il quale si confornta il valore della serie.
Tale funzione descrive quindi il modo in cui le osservazioni passate influenzano quelle future: valori positivi indicano una correlazione positiva tra X_{t_i} e X_{t_i+l}, mentre valori negativi suggeriscono una relazione inversa.
Si osservi che la funzione di autocovarianza è pari, ovvero $\gamma(l) = \gamma(-l)$, poiché la covarianza è una misura commutativa: $\mathrm{Cov}(X_{t_i+l}, X_{t_i}) = \mathrm{Cov}(X_{t_i}, X_{t_i+l})$. Inoltre, il valore dell'autocovarianza per il ritardo nullo corrisponde alla varianza:

$$\gamma_{t_i}(0) = \mathrm{Cov}(X_{t_i}, X_{t_i}) = \mathrm{Var}(X_{t_i})$$

Nel caso di processi stazionari, l'autocovarianza dipende unicamente dal lag l e non dall'istante specifico t_i: si indica con $\gamma(l)$.

Funzione di Autocorrelazione (ACF)

L'ACF misura la correlazione tra un'osservazione della serie e le sue osservazioni ritardate di un certo numero di periodi, ovvero, mostra quanto la serie temporale è correlata con se stessa a diversi lag temporali.
Ciò vuol dire che un picco significativo nel grafico dell'ACF in corrispondenza di un certo lag l indica la presenza di una forte correlazione tra le osservazioni separate da l step temporali. Invece, un decadimento graduale dei valori dell'ACF verso zero, suggerisce una dipendenza a lungo termine oppure la presenza di un trend nella serie.
Formalmente, l'autocorrelazione della variabile X_{t_i} al lag l è definita come:

$$\rho_{t_i}(l) = \frac{\mathrm{Cov}(X_{t_i}, X_{t_i+l})}{\sqrt{\mathrm{Var}(X_{t_i})\,\mathrm{Var}(X_{t_i+l})}}$$

Nel caso di una serie stazionaria, la covarianza è indipendente dal tempo e la varianza $\mathrm{Var}(X_{t_i}) = \gamma(0)$ è costante nel tempo, per cui il coefficiente di autocorrelazione dipende esclusivamente dal lag l:

$$\rho(l) = \frac{\mathrm{Cov}(X_t, X_{t+l})}{\mathrm{Var}(X_t)} = \frac{\gamma(l)}{\gamma(0)}$$

dove t è un istante di tempo qualsiasi.

Funzione di Autocorrelazione Parziale (PACF)

La PACF misura la correlazione tra un'osservazione e una sua osservazione ritardata di lag l, eliminando l'effetto dei ritardi intermedi. Rappresenta cioè la correlazione "pura" tra X_t e X_{t+l} una volta rimossa l'influenza delle osservazioni comprese tra di esse.
Formalmente, l'autocorrelazione parziale al lag l è definita come:

$$\psi_{t_i}(l) = \frac{\mathrm{Cov}\big(X_{t_i}, X_{t_i+l} \,|\, X_{t_i+1}, \ldots, X_{t_i+l-1}\big)}{\sqrt{\mathrm{Var}\big(X_{t_i} \,|\, X_{t_i+1}, \ldots, X_{t_i+l-1}\big)\,\mathrm{Var}\big(X_{t_i+l} \,|\, X_{t_i+1}, \ldots, X_{t_i+l-1}\big)}}$$

Analogamente, se la serie è stazionaria, l'espressione si semplifica poiché non vi è più dipendenza esplicita dall'istante temporale t_i:

$$\psi(l) = \frac{\mathrm{Cov}(X_t, X_{t+l} \mid X_{t+1}, \ldots, X_{t+l-1})}{\mathrm{Var}(X_t \mid X_{t+1}, \ldots, X_{t+l-1})}$$

2.2 Metriche utilizzate

Siccome verrano testati due diversi modelli con lo scopo di eseguire un forecast, sono necessarie delle metriche per valutarne l'accuratezza in fase di analisi e per confrontare le prestazioni dei modelli tra di loro.
Esistono numerose metriche di errore, ciascuna con caratteristiche diverse che permettono di attribuire maggiore rilevanza a certi aspetti piuttosto che ad altri, come la scala dei dati, la presenza di outliers o la variabilità della serie.
Nel presente lavoro si considereranno i seguenti indicatori: il Mean Absolute Error (MAE), il Mean Absolute Percentage Error (MAPE) e il Root Mean Squared Scaled Error (RMSSE) e il Continuous Ranked Probability Score (CRPS). Infine, verrà presentato l'Akaike Information Criterion (AIC), utile per confrontare la capacità di diversi tipi di modelli SARIMA di adattarsi ai dati.

2.2.1 Mean Absolute Error

Il MAE misura l'errore medio assoluto tra i valori previsti dal modello e quelli effettivi. Si calcola come segue:

$$MAE = \frac{1}{T}\sum_{t=1}^{T} |x_t - \hat{x}_t|,$$

dove x_t è il valore vero della serie, $\hat{x}_t$ è il valore predetto dal modello al tempo t e T è il numero totale di osservazioni su cui ci si basa.
Il MAE è un tipo di metrica semplice e facilmente interpretabile perché mantiene la stessa unità di misura dei dati, ma allo stesso tempo è influenzato dalla grandezza assoluta della serie. Inoltre non è sensibile alla direzione dell'errore (positivo o negativo) per la presenza del valore assoluto.
Infine, ha il vantaggio di penalizzare in modo lineare gli errori ed è quindi robusto rispetto agli outlier, al contrario di metriche come ad esempio il Mean Squared Error (MSE).

2.2.2 Mean Absolute Percentage Error

Il MAPE esprime l'errore medio in percentuale, rispetto ai valori osservati cioè indica in media quanto il valore previsto si discosta da quello osservato in termini percentuali. Si misura come segue:

$$MAPE = \frac{1}{T}\sum_{t=1}^{T} \left|\frac{x_t - \hat{x}_t}{x_t}\right|.$$

Il MAPE permette quindi di confrontare modelli caratterizzati da scale diverse ed è facilmente interpretabile.
Il maggiore limite è che la sua definizione vale solo per le osservazioni con valore diverso da zero, di conseguenza tende a sovrastimare gli errori su valori piccoli e sottostimare quelli su valori elevati.
Nel caso specifico dei dati utilizzati in questa trattazione non sorge questo problema poiché i valori medi si attestano intorno ai 225 $\mu mol/l$.

2.2.3 Root Mean Squared Scaled Error

Il RMSSE è la versione scalata Root Mean Squared Error (RMSE), nato per tenere in considerazione la variabilità della serie.
È stato utilizzato, ad esempio, nel contesto della Accuracy Challenge nella competizione M5 [8] di Kaggle nel 2020, con l'obiettivo di valutare le performance del forecasting dei modelli proposti dai partecipanti. Si calcola come segue, denotando con n la dimensione del training sample e con h la lunghezza del periodo di forecast:

$$RMSSE = \sqrt{\frac{\frac{1}{h}\sum_{t=n+1}^{n+h}(x_t - \hat{x}_t)^2}{\frac{1}{n-1}\sum_{t=2}^{n}(x_t - x_{t-1})^2}}.$$

Il numeratore rappresenta l'errore quadratico medio (MSE) tra il valore vero e quello previsto; il denominatore invece normalizza tale errore rispetto alla variabilità intrinseca della serie, misurata tramite le differenze di lag pari a uno dei suoi valori effettivi. Il rapporto tra i due membri rende adimensionale il valore della metrica.
La scelta del denominatore risulta in un errore ridotto nel caso di una serie molto variabile nel train set (si accetta una minore precisione) e in un errore più elevato nel caso di una serie meno variabile (l'errore viene amplificato in quanto si richiede maggiore precisione).
A causa del quadrato applicato, l'RMSSE ha come svantaggio il fatto di essere sensibile agli outliers.
Tuttavia, il denominatore di tale metrica è stato formulato in questo modo perché i partecipanti della competizione M5 non avevano accesso ai veri dati di test, e dovevano basarsi sui dati di addestramento e sulla previsione della serie. Invece, nel contesto di questa trattazione, i dati di test sono sempre disponibili, nel senso che è possibile scegliere una finestra temporale di addestramento alla quale seguono altri dati da confrontare con la previsione. Per questo motivo, si procede con la modifica (m) del denominatore della metrica, trasferendo la valutazione dal periodo di addestramento al periodo di test.

$$mRMSSE = \sqrt{\frac{\frac{1}{h}\sum_{t=n+1}^{n+h}(x_t - \hat{x}_t)^2}{\frac{1}{h}\sum_{t=n+1}^{n+h}(x_t - x_{t-1})^2}}.$$

In questo modo, dei valori prossimi a 1 indicano che il modello prevede con la stessa accuratezza di una previsione col metodo Naive, cioè che utilizza x_{t-1} come previsione di

x_t; valori minori di 1 indicano un modello migliore della previsione Naive, mentre valori maggiori di 1 indicano prestazioni peggiori.
Tuttavia, nel caso delle previsioni multi-step, questo non è del tutto vero: un modello Naive prevederebbe lo stesso valore, ovvero l'ultimo dato del training set, per tutto l'arco temporale del test.
Diversamente, la metrica rispetterebbe tale interpretazione se il modello che si sta valutando inglobasse nel training set del passo successivo la previsione one-step che effettua al passo corrente.
Per questa trattazione si riporteranno comunque i valori del mRMSSE ottenuti, tenendo conto delle considerazioni precedenti.

2.2.4 Continuous Ranked Probability Score

Il CRPS è una metrica di valutazione utilizzata per misurare l'accuratezza delle previsioni di tipo probabilistico. Sono quindi adatte per valutare modelli che non restituiscono delle stime puntuali, bensì delle distribuzioni di probabilità. In questo senso si può interpretare come una generalizzazione del MAE [11].
La sua formulazione, considerando un singolo step e dato y il valore reale della serie, è la seguente:

$$\mathrm{CRPS}(F, y) = \int_{-\infty}^{+\infty} [F(x) - H(x \geq y)]^2 dx,$$

dove:

- $F(x) = \mathbb{P}(X \leq x)$ è la funzione di ripartizione (o funzione cumulativa) della variabile X predetta dal mdoello;
- $H(x \geq y) = \begin{cases} 1 & \text{se } x \geq y \\ 0 & \text{se } x < y \end{cases}$ è la funzione di Heaviside, che corrisponde alla funzione di ripartizione delle singole osservazioni. Rappresenta la probabilità che i valori osservati siano al di sotto di una data soglia.

La differenza tra i due termini misura quanto la funzione di ripartizione predetta si discosta da quella effettiva, mentre l'elevamento a quadrato ha l'obiettivo di penalizzare maggiormente le discrepanze elevate rispetto a quelle più contenute. Infine, l'integrale fornisce una misura globale di tali discrepanze.
L'integrale si può scrivere in forma chiusa nel caso di distribuzioni predette di tipo gaussiano [4], come nel caso del modello SARIMA. La formulazione diventa la seguente:

$$\mathrm{CRPS}(\mathcal{N}(\mu, \sigma^2), y) = \sigma \cdot \left[\frac{1}{\sqrt{\pi}} - 2\,\phi\left(\frac{y-\mu}{\sigma}\right) - \frac{y-\mu}{\sigma}\left(2\,\Phi\left(\frac{y-\mu}{\sigma}\right) - 1\right)\right]$$

dove:

- $\phi(\cdot)$ è la densità della normale standard $\mathcal{N}(0,1)$;
- $\Phi(\cdot)$ è la funzione di distribuzione cumulativa della normale standard $\mathcal{N}(0,1)$.

Il termine σ che moltiplica deriva dal cambio di variabili $z = \frac{x-\mu}{\sigma} \Rightarrow dz = \frac{dy}{\sigma} \Rightarrow dy = \sigma dz$ eseguito, e indica la dispersione della distribuzione. Se la distribuzione è più larga (σ grande), il CRPS sarà proporzionalmente più grande anche in presenza di errori contenuti, per penalizzare l'incertezza della previsione. Se la distribuzione è più stretta (σ piccolo), il CRPS sarà più piccolo, a parità di scostamento.

Si anticipa che tale metrica sarà calcolata solo rispetto ai risultati del moello SARIMA, che fornisce previsioni probabilistiche.

2.2.5 Akaike Information Criterion

La tecnica utilizzata in questo lavoro per confrontare tra loro i modelli ARIMA con diversi parametri, è quella dell'AIC (Akaike Information Criterion), un criterio che tiene conto sia del numero di parametri stimati (k), sia della bontà di adattamento del modello tramilte la massima verosimiglianza (L): l'obiettivo è quello di trovare un trade-off tra accuratezza e complessità.
Il punteggio AIC viene calcolato come segue:

$$\mathrm{AIC} = 2k - 2\log L$$

Il termine $2k$ rappresenta quindi una sorta di penalità, in quanto un numero elevato di parametri potrebbe generare overfitting, ovvero un adattamento eccessivo ai dati di addestramento che compromette la capacità di generalizzare su nuovi dati.
Se la differenza tra gli AIC dei modelli è circa uguale a 2 unità, allora possono essere considerati equivalentemente validi, altrimenti il modello da preferire è quello con punteggio AIC minore, poiché rappresenta i dati in modo efficace con un numero limitato di parametri.

Capitolo 3

Il Modello SARIMA

Per introdurre il modello SARIMA, è necessario innanzitutto chiarire alcuni concetti che verranno utilizzati in seguito [9]:

- Operatore Backshift B: detto anche operatore di ritardo, rappresenta un ritardo temporale o uno spostamento all'indietro nei dati della serie temporale. Nello specifico, se x_t rappresenta un valore della serie temporale al tempo t, l'operatore backshift applicato a x_t è rappresentato come Bx_t, e indica il valore della serie temporale al tempo precedente, cioè:

  $$Bx_t = x_{t-1}$$

  L'applicazione ripetuta n volte di B viene indicata con la scrittura B^n, quindi si ha:

  $$B^n x_t = x_{t-n}$$

- Rumore Bianco (White Noise): è il processo stocastico più semplice, il "blocco costruttivo di base" [5], e verrà indicato con ω_t. Si tratta di una sequenza di variabili casuali di media nulla e varianza pari a σ^2, indipendenti e identicamente distribuite (i.i.d.): non è quindi presente alcuna correlazione con le osservazioni precedenti o successive. Infatti l'autocorrelazione (parziale e totale) è nulla per ogni lag temporale diverso da zero.
  Un processo che soddisfa queste condizioni e che segue la distribuzione di una normale, è detto Rumore Bianco Gaussiano:

  $$x_t = \omega_t, \qquad \omega_t \sim \mathcal{N}(0, \sigma^2)$$

  Nella modellazione, il rumore bianco rappresenta la componente puramente casuale non spiegata dal modello. Se i residui di un modello si comportano come un rumore bianco, il modello può essere considerato una buona rappresentazione dei dati osservati.

- Integrazione di ordine d: è un'operazione utile per rendere stazionarie serie temporali che non lo sono, ad esempio per poter utilizzare i modelli ARIMA. Consiste nel

calcolare le differenze successive tra i valori osservati.
Ad esempio, la differenza prima di una serie è definita come:

$$y_t = x_t - x_{t-1} = x_t - Bx_t = (1 - B)x_t$$

Mentre una differenziazione di ordine d è data da:

$$y_t = (1 - B)^d x_t$$

Allo stesso modo, è possibile effettuare una differenziazione stagionale, sottraendo a ciascuna osservazione quella corrispondente al lag stagionale precedente, cioè:

$$y_t = x_t - x_{t-s} = (1 - B^s)x_t$$

Una serie si dice integrata di ordine d se, dopo d differenziazioni, diventa un rumore bianco, ovvero:

$$(1 - B)^d x_t = \omega_t$$

A questo punto è possibile iniziare con lo studio dei modelli per la rappresentazione e la previsione delle serie storiche. Si procederà per step, analizzando prima il modello Autogregressivo (AR), poi il modello a Media Mobile o Moving Average (MA) e infine il risultato della loro combinazione.

3.1 Modello Autoregressivo AR(p)

I modelli autoregressivi di ordine p rappresentano i dati in questo modo:

$$x_t = \alpha_1 \cdot x_{t-1} + \alpha_2 \cdot x_{t-2} + ... + \alpha_p \cdot x_{t-p} + w_t$$

cioè, il valore della serie al tempo t è una combinazione lineare dei propri valori passati, più un termine di white noise. La presenza di white noise si può interpretare come un "disturbo" a un modello di regressione.
L'ordine p indica da quante osservazioni precedenti dipende il dato attuale (al tempo t), mentre il valore dei coefficienti α misurano la forza della dipendenza. Il processo è markoviano di ordine p, ovvero il valore futuro della serie dipende esclusivamente dalle p realizzazioni passate: di conseguenza l'autocorrelazione parziale si annullerà per lag maggiori di p.
Tramite l'utilizzo dell'operatore backshift, è possibile scrivere il modello in forma compatta:

$$\begin{aligned}(1 - \alpha_1 B - \alpha_2 B^2 - ... - \alpha_p B^p)x_t &= \omega_t \\ \theta_p(B)x_t &= \omega_t\end{aligned}$$

dove $\theta_p(B)$ rappresenta il polinomio autoregressivo in funzione di B e di grado p.

3.2 Modello Moving Average MA(q)

I modelli a media mobile di ordine q sono definiti come:

$$x_t = \omega_t + \beta_1 \cdot \omega_{t-1} + \beta_2 \cdot \omega_{t-2} + ... + \beta_q \cdot \omega_{t-q}$$

Questo modello prevede che esista dipendenza dai rumori bianchi passati, ovvero gli errori commessi durante l'adattamento ai dati in precedenza, e non dalle osservazioni passate, come previsto nel modello precedente.
L'ordine q indica quanti rumori bianchi passati è necessario considerare e β quanto è forte la dipendenza. Per i modelli Moving Average, l'autocorrelazione decade a 0 per lag temporali maggiori di q.
Usando l'operatore backshift, si può scrivere il modello come segue:

$$\begin{aligned} x_t &= (1 + \beta_1 B + \beta_2 B^2 + ... + \beta_q B^q)\omega_t \\ x_t &= \phi_q(B)\omega_t \end{aligned}$$

dove $\phi_q(B)$ rappresenta il polinomio in funzione di B di ordine q.

Combinando i modelli AR(p) e MA(q) si ottiene il modello ARMA(p, q), che permette di scrivere la serie come:

$$\begin{aligned} x_t &= \alpha_1 x_{t-1} + ... + \alpha_p x_{t-p} + \omega_t + \beta_1 \omega_{t-1} + ... + \beta_q \omega_{t-q} \\ \theta_p(B)x_t &= \phi_q(B)\omega_t \end{aligned}$$

Infine, se la serie non è stazionaria, si applica una differenziazione di ordine d fino ad ottenere il risultato desiderato, per poi applicare un modello ARMA. SI ottiene così un modello ARIMA di parametri (p, d, q), che si scrive in forma compatta come segue:

$$\theta_p(B)(1 - B)^d x_t = \phi_q(B)\omega_t$$

3.3 Modello SARIMA $(p, d, q)(P, D, Q)_s$

Nei modelli presentati fino ad ora, non viene considerata la componente stagionale della serie. I modelli SARIMA$(p, d, q)(P, D, Q)_s$ sono una naturale estensione dei modelli ARIMA(p, d, q), arricchiti dalla la capacità di gestire pattern stagionali.
I primi tre parametri che li rappresentano, (p, d, q), sono gli ordini dei modelli AR, MA e I rispettivamente, mentre (P, D, Q) sono gli ordini dei modelli AR, MA e I stagionali. Il termine s indica la lunghezza del periodo stagionale.
Si ottiene un modello del tipo:

$$\Theta_P(B^s)\theta_p(B)(1 - B^s)^D(1 - B)^d x_t = \Phi_Q(B^s)\phi_q(B)\omega_t$$

Questa formulazione permette di combinare con flessibilità le dinamiche a breve termine e i pattern stagionali, requisito che risulta fodamentale per lo studio delle serie storiche.

Nel risvolto applicativo di questa trattazione, sviluppato nella sezione 3.5, verrà utilizzato il modello `SARIMAX` della libreria `statsmodels` [14], che permette l'inclusione della stagionalità (S) e dei regressori, o variabili esogene (X): quei fattori esterni significativi che possono influenzare il modello e le previsioni.
Nello specifico, gli ordini del modello SARIMA possono essere indicati tramite una tupla di tre numeri nel parametro `order`, così come per gli ordini stagionali nel parametro `seasonal_order`. Il dataframe o la serie contentente le variabili esogene, a condizione che ci sia conformità tra le date delle misurazioni della variabile principale, può essere passato tramite il parametro `exog`.

3.4 Stime dei parametri e dell'incertezza

Si consideri per semplicità una serie stazionaria (per esempio già differenziata, se necessario) e di media nulla (per esempio ottenuta sottraendo la media della serie alla serie stessa) descritta da un modello ARMA(p, q). Il caso stagionale è analogo.
Tale serie si può scrivere come:

$$\begin{aligned} x_t &= \alpha_1 x_{t-1} + ... + \alpha_p x_{t-p} + \omega_t + \beta_1 \omega_{t-1} + ... + \beta_q \omega_{t-q}, && \omega_t \sim \mathcal{N}(0, \sigma^2) \\ &= \underbrace{\sum_{i=1}^{p} \alpha_i x_{t-i} + \sum_{j=i}^{q} \beta_i \omega_{t-i}}_{\mu_t} + \omega_t, && \omega_t \sim \mathcal{N}(0, \sigma^2) \end{aligned}$$

Si può dunque derivare il seguente risultato, che verrà anche sfruttato in fase di previsione:

$$x_t \mid x_{t-1}, \ldots, x_{t-p} \sim \mathcal{N}(\mu_t, \sigma^2) \tag{3.1}$$

Una volta dedotti gli ordini (p, d, q) del modello, e quelli stagionali (P, D, Q) nel caso di un modello SARIMA, è necessario stimare i coefficienti α_i, β_i delle componenti autoregressive e di moving average e la varianza del rumore bianco. Il vettore dei parametri da stimare è $\boldsymbol{\theta} = \{\alpha_1, \ldots, \alpha_p, \sigma^2, \beta_1, \ldots, \beta_q\}$.
La tecnica utilizzata è quella della massimizzazione della verosimiglianza ℓ. L'idea di questo metodo consiste nello scegliere, tra tutti i possibili valori dei parametri $\boldsymbol{\theta}$, quelli che rendono massima la probabilità di osservare la serie temporale effettiva, i dati $\mathbf{x} = \{x_1, \ldots, x_T\}$. Si vuole quindi massimizzare la probabilità congiunta dei dati:

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} f(\mathbf{x} \mid \boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} f(\boldsymbol{\theta} \mid \mathbf{x}) = \arg\max_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}).$$

Tale massimizzazione, a causa della non linearità della funzione di log-verosimiglianza, viene eseguita tramite metodi iterativi numerici. In particolare, la libreria `statsmodels` permette di scegliere il metodo di ottimizzazione da utilizzare, specificandolo nel parametro `method` all'interno di `fit()`. I valori dei parametri stimati possono essere recuperati salvando il risultato dell'adattamento del moello in un oggetto, ad esempio `result`, e poi essere richiamati con l'attributo `.params`.

Per quanto riguarda la previsione e la sua incertezza, il modello ARMA utilizza la formula ricorsiva 3.1, quindi:

$$x_{t+1} = \sum_{i=1}^{p} \hat{\alpha}_i x_{t+1-i} + \sum_{j=i}^{q} \hat{\beta}_i \omega_{t+1-i} + \omega_{t+1}, \qquad \omega_{t+1} \sim \mathcal{N}(0, \hat{\sigma}^2)$$

Il problema è che il termine casuale futuro ω_{t+1} non è osservabile, quindi il valore che si può ottenere data tutta l'informazione $\mathcal{F}_t$ fino al tempo t è una media:

$$\mathbb{E}[x_{t+1} \mid \mathcal{F}_t] = \sum_{i=1}^{p} \hat{\alpha}_i x_{t+1-i} + \sum_{j=i}^{q} \hat{\beta}_i \hat{\omega}_{t+1-i} = \hat{\mu}_{t+1}$$

La previsione è una variabile aleatoria data dalla somma tra la media (deterministica) appena stimata e il rumore bianco (aleatorio):

$$x_{t+1} = \hat{\mu}_{t+1} + \omega_{t+1}, \qquad \omega_{t+1} \sim \mathcal{N}(0, \hat{\sigma}^2)$$

Gli intervalli di confidenza relativi a tale previsione si basano quindi sulla varianza stimata a partire dai valori passati della serie. In particolare, i due estremi $l_{1,2}$ degli intervalli di livello $(1-\alpha)$ sono calcolati come segue, a partire dalla stima $\hat{x}_{t+1} \mid \mathcal{F}_t \sim \mathcal{N}(\hat{\mu}_{t+1}, \hat{\sigma}^2)$.

$$l_{1,2} = \hat{\mu}_{t+1} \pm z_{1-\frac{\alpha}{2}} \cdot \hat{\sigma}$$

dove $z_{1-\frac{\alpha}{2}}$ è il quantile della distribuzione normale standard associata a un livello di confidenza pari a $1-\alpha$: ad esempio, $z_{0.975} = 1.96$ per il valore di default $\alpha = 0.05$.
Questo metodo, però, fornisce gli intervalli di previsione "condizionali" [5], nel senso l'incertezza considerata è dovuta solo alla variabilità della serie (la varianza della gaussiana), e alla sua propagazione negli step futuri. Tali intervalli non tengono conto della variabilità dovuta alle stime dei parametri $\hat{\boldsymbol{\theta}}$, da cui derivano a loro volta le stime $\hat{\mu}_{t+1}$ e $\hat{\sigma}$, bensì questi termini vengono considerati noti. La variabilità totale che in realtà comporta ogni previsione è data dalla somma dei due contributi:

$$\mathrm{Var}(x_{t+1}) = \underbrace{\mathrm{Var}(x_{t+1} \mid \boldsymbol{\theta})}_{\text{incertezza condizionale}} + \underbrace{\mathrm{Var}_{\boldsymbol{\theta}}(\mathbb{E}[x_{t+1} \mid \boldsymbol{\theta}])}_{\text{incertezza parametrica}}$$

Nel caso di previsioni a più passi in avanti (multi-step), l'incertezza del modello tende ad aumentare con l'orizzonte temporale della previsione. Infatti, ogni nuova stima dipende dalle previsioni precedenti, che a loro volta contengono un margine di errore: in questo modo la varianza si propaga e si somma progressivamente ad ogni passo futuro. Di conseguenza, gli intervalli di confidenza diventano più ampi man mano che ci si allontana dal primo istante del forecast.

3.5 Applicazione del modello SARIMA

In questa sezione verranno presentati i risultati ottenuti nella fase di adattamento e di previsione applicando un modello SARIMA, oltre che la metodologia seguita per capire i parametri adatti da assegnare alla componente autoregressiva e quella a media mobile.

3.5.1 Analisi preliminari

Come prima cosa, per poter adattare un modello SARIMA, è necessario mostrare che la serie sia stazionaria, o provvedere ad eseguire tecniche che la rendano tale.
Considerando l'intero arco temporale a disposizione, si riporta il grafico relativo all'ACF e alla PACF fino al lag 48 della serie dell'ossigeno disciolto:

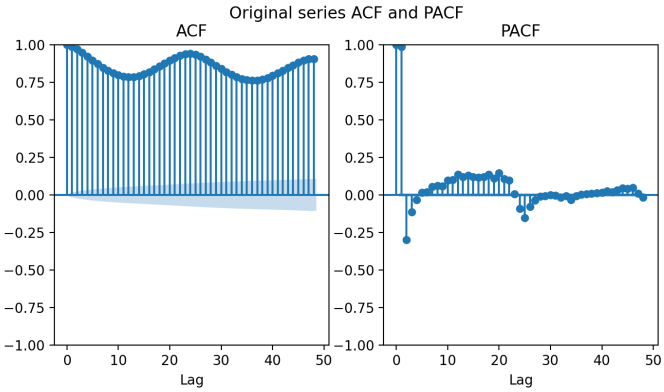


Figura 3.1: ACF e PACF della serie originale.

Dall'analisi del grafico dell'ACF si nota un lento decadimento della correlazione, che oscilla con un periodo regolare di 24 lag. Questo comportamento indica una forte non stazionarietà, perché l'autocrrelazione non si riduce in modo rapido a valori prossimi a zero. Inoltre suggerisce una componente stagionale di periodo 24, come ragionevolmente ci si aspetta.
Il grafico della PACF, invece, presenta 2 picchi significativi (i primi due) e un ulteriore picco (il terzo) di rilevanza minore ma non del tutto trascurabile, seguiti da un rapido decadimento. Questo suggerisce la presenza di una componente autoregressiva (p) di ordine pari a 1, 2 o 3.

Per rendere la serie stazionaria, si procede ad eseguire una differenziazione stagionale di periodo 24. La serie appare così:

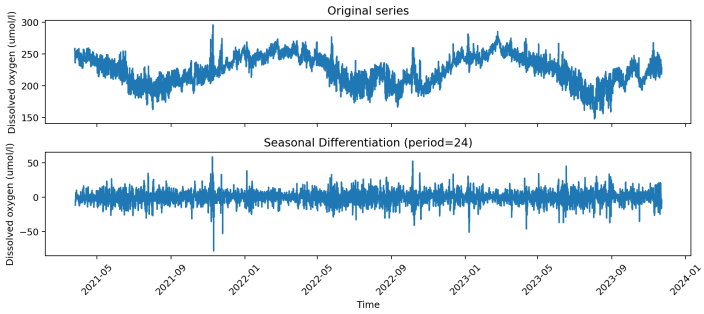


Figura 3.2: Confronto tra serie originale e serie differenziata stagionalmente con periodo 24.

Per confermare la stazionarietà della serie, si procede con un Augmented Dickey-Fuller test [7], la cui ipotesi nulla prevede la presenza di una radice unitaria, ossia la non stazionarietà della serie. Il test, eseguito sulla serie differenziata, fornisce questo output:

```
(np.float64(-28.968519078995648),
 0.0,
 47,
 23254,
 {'1%': np.float64(-3.4306312428829964),
  '5%': np.float64(-2.8616643004278943),
  '10%': np.float64(-2.5668361615548476)},
 np.float64(121068.09427822738))
```

I valori ottenuti, in ordine, rappresentano:

- il valore della statistica di test di Dickey-Fuller: più è negativo, più è forte l'evidenza contro l'ipotesi nulla. in questo caso il valore è molto negativo, e quindi la serie è stazionaria.
- il p-value associato alla statistica, ovvero la probabilità di osservare un valore di test così estremo se la serie non fosse stazionaria. Si deduce ancora una volta la forte stazionarietà della serie.
- il numero n di lag utilizzati nel modello ADF.
- il numero di osservazioni su cui si basa il test, dopo aver rimosso i primi n lag.
- i tre valori critici corrispondenti ai tre diversi livelli di significatività (1%, 5%, 10%). In particolare, se il valore della statistica di test è minore del valore critico, allora la stazionarietà è confermata con una confidenza pari al livvelo di significatività corrispondente. In questo caso, approssimando, -28 è molto minore del primo valore critico, ovvero -3, (e ragionevolmente anche degli altri due), quindi il rifiuto dell'ipotesi nulla avviene con elevatissima sicurezza.
- il punteggio AIC del modello ADF. In questo caso non è informativo, perché non c'è un confronto tra diversi modelli.

A questo punto, analizzando i grafici relativi ad ACF e PACF della serie differenziata, si ottiene:

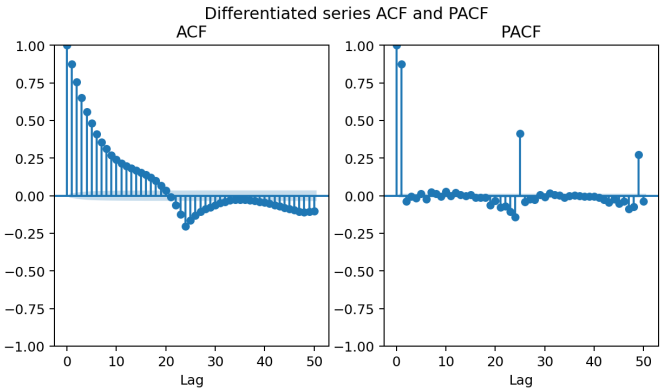


Figura 3.3: ACF e PACF della serie differenziata stagionalmente con periodo 24.

Dopo la differenziazione stagionale, l'andamento delle funzioni di autocorrelazione (ACF) e autocorrelazione parziale (PACF) risulta significativamente modificato, e rappresenta un punto di partenza utile per dedurre gli ordini del modello SARIMA [11].
Per quanto riguarda l'ACF, si nota innanzitutto l'assenza dell'andamento ondulatorio che caratterizzava il grafico precedente. I valori adesso risultano decadere verso lo zero: tale evidenza riconferma che la componente stagionale è stata correttamente rimossa e che la serie differenziata risulta stazionaria.
Inoltre l'assenza di picchi significativi isolati non sembra suggerire un ordine specifico per quanto rioguarda l'eventuale presenza di una componente moving average. Tuttavia, nei primi lag si hanno valori di autocorrelazione abbastanza elevati, e quindi non si può escluderne la presenza.
La PACF invece presenta un decadimento netto dopo i primi due lag, che conferma la componente autoregressiva di basso ordine intuita in precedenza. Sono presenti poi degli spike isolati che decrescono di volta in volta ai lag multipli di 24, segno di una componente autoregressiva di tipo stagionale; questo ci suggerisce di tentare valori del parametro P diversi da zero.

3.5.2 Implementazione del modello

Per la scelta dei parametri $(p, d, q)(P, D, Q)$ del modello, è stata condotta una ricecra mediante cicli annidati. In particolare, per quanto dedotto dall'analisi preliminare dei grafici di ACF e PACF, sono state definite le seguenti combinazioni di ordini da testare:

- il parametro s relativo alla stagionalità è fissato a 24;
- gli ordini di differenziazione d e D sono stati variati tra 0 e 1, tuttavia la scelta $D = 1$ si era già rivelata necessaria per rendere la serie stazionaria;
- gli ordini non stagionali p e q sono stati fatti variare tra 0 e 3;

- gli ordini stagionali P e Q sono stati fatti variare tra 0 e 2.

A causa dell'elevata complessità computazionale associata alla ricerca di tali combinazioni, il periodo di addestramento è stato inizialmente limitato a un mese di dati, per consentire una prima valutazione rapida dei parametri. Successivamente, la finestra di training è stata progressivamente estesa a sei mesi e infine a un intero anno, al fine di valutare come la quantità di dati storici influenzasse le prestazioni del modello e la stabilità dei parametri stimati.
Per ciascuna combinazione e per ciascun periodo è stato calcolato il punteggio AIC (trattato nella sezione 2.2.5): i risultati sono stati ordinati per punteggio decrescente, e le prime righe delle tabelle, corrispondenti ai modelli migliori, sono riportate di seguito:

$(p,d,q)(P,D,Q)$	AIC
(2,1,1)(1,1,2)	3124.56
(3,1,3)(1,1,2)	3124.90
(1,1,3)(1,1,1)	3129.10
(1,1,3)(0,1,2)	3129.16
(3,1,3)(1,1,1)	3129.65

Tabella 3.1: 1 mese di train

$(p,d,q)(P,D,Q)$	AIC
(3,1,2)(1,1,2)	21661.34
(2,1,3)(1,1,2)	21663.32
(2,1,2)(1,1,2)	21664.20
(3,1,2)(2,1,2)	21665.79
(2,1,2)(2,1,2)	21677.18

Tabella 3.2: 6 mesi di train

$(p,d,q)(P,D,Q)$	AIC
(2,1,2)(1,1,2)	42250.80
(3,1,2)(1,1,2)	42252.99
(2,1,3)(1,1,2)	42259.08
(2,1,3)(2,1,2)	42276.68
(3,1,2)(2,1,1)	42282.14

Tabella 3.3: 12 mesi di train

Tabella 3.4: Confronto dei valori AIC per diversi gruppi di configurazioni degli oridni del modello SARIMA.

Dalle tabelle si nota innanzitutto una coerenza negli ordini stagionali risultati migliori: ai primi posti compare sempre la configurazione $(P,D,Q) = (1,1,2)$.
Per quanto riguarda gli ordini non stagionali, appare una certa variabilità: attribuendo maggiore rilevanza ai risultati ottenuti con più mesi di addestramento, p e q risultano sempre pari o maggiori di 2. Il grado di differenziazione d, invece, è sempre pari a 1.
Nel complesso, considerando che le configurazioni con AIC più bassi presentano strutture simili, e che le differenze di punteggio sono significative se superiori alla soglia di 2 punti, la configurazione migliore dei parametri risulta essere SARIMA $(2,1,2)(1,1,2)_{24}$.
Per confermare tale struttura, si procederà con un'ulteriore verifica. Sempre utilizzando un anno di dati come periodo di addestramento del modello e impostando il parametro $s = 24$, sono state esplorate queste configurazioni:

- gli ordini di differenziazione, siccome nei risultati precdenti sono sempre stati pari a 1 e mai nulli, sono stati fatti variare tra 1 e 2;
- gli ordini non stagionali p e q, che sono comparsi nelle prime posizioni sempre pari o maggiori di 2, sono stati fatti variare tra 2 e 3;
- l'ordine stagionale P è stato fissato a 1, perché è comparso sempre pari a 1, nonostante potesse assumere anche il valore di 2;
- l'ordine stagionale Q, siccome nella maggior parte dei casi migliori era pari a 2, è stato fatto variare tra 2 e 3.

I risultati migliori emersi da questa prova sono riportati di seguito, sempre in ordine decrescente di punteggio AIC:

$(p,d,q)(P,D,Q)$	AIC
(2, 1, 2)(1, 1, 2)	42250.80
(3, 1, 2)(1, 1, 2)	42252.99
(2, 1, 3)(1, 1, 2)	42259.08
(2, 1, 3)(1, 1, 3)	42270.85
(3, 1, 3)(1, 1, 2)	42276.52

Tabella 3.5: Confronto tra modelli ARIMA stagionali e loro AIC.

Il "podio" di tali risultati è il medesimo della tabella 3.4, confermando che il modello che meglio si adatta ai dati dell'ossigeno è il SARIMA(2,1,2)(1,1,2)$_{24}$.

Per valutare le prestazioni del modello SARIMA individuato, e confrontare le sue previsioni con i dati reali, è stata implementata una funzione denominata `fit_compare_sarimax`.

```
def fit_compare_sarimax(ox_serie, order, sorder, s = 24, start_train =
    None, start_forecast = None, days_to_predict = 3, exog = None, plot
    = True):
```

Tale funzione consente di stimare un modello SARIMA o SARIMAX (nel caso in cui vengano fornite variabili esogene, passabili tramite il parametro `exog`) su un intervallo temporale di training, di effettuare una previsione per un determinato orizzonte temporale successivo e di visualizzare graficamente i risultati. In particolare, la funzione riceve in input la serie temporale di interesse sotto forma di Pandas Series, gli ordini del modello non stagionali (p,d,q) e quelli stagionali (P,D,Q). La stagionalità è impostata a 24 di default, siccome deve catturare una dinamica giornaliera.
La funzione suddivide sia la serie che il dataframe delle variabili esogene in 2 parti: una per l'addestramento (train set) e una per la previsione (test set), sulla base delle date specificate in input (`start_train`, `start_forecast`) e sul numero di giorni da prevedere (`days_to_predict`), di default pari a 3 giorni, ovvero 72 step orari. Se le due date non vengono fornite, la funzione memorizza il primo indice temporale del dataframe in input come data di inizio del training, e l'indice finale traslato indietro di tre giorni come data iniziale della previsione.
In seguito viene creato il modello chiamando la funzione `SARIMAX` di `statsmodels`, il risultato del fit viene salvato in un oggetto e infine viene estratta la previsione desiderata. Inoltre si concatenano i valori stimati durante il fit della serie nella fase di training con quelli della previsione, e la serie ottenuta viene restituita dalla funzione.
Per visualizzare graficamente i risultati, viene stampato il grafico che confronta l'intera curva stimata (in blu), che comprende fit del modello e previsione, con quella osservata (rappresentata dai pallini neri nel train set e da una curva verde nel test set). In azzurro sono invece rappresentati gli intervalli di confidenza stimati dal modello.

Per mostrare i risultati della funzione `fit_compare_sarimax`, verrà utilizzato un periodo di un mese come training set al fine di stimare il modello, e saranno generate previsioni sui tre giorni successivi. In particolare si sceglie il mese di marzo 2022, quindi si imposta:

- `start_train = '2022-03-01 00:00:00'`
- `start_forecast = '2022-04-01 00:00:00'`

Il risultato del modello univariato su questo periodo è il seguente:

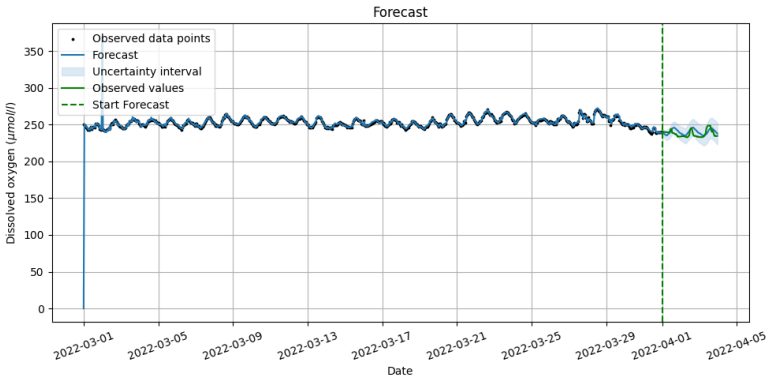


Figura 3.4: Confronto tra valori reali e previsti dal modello SARIMA univariato.

La prima cosa che si nota è che i valori iniziali, risultato dell'adattamento del modello ai dati di train, si discostano molto dalla realtà e sono prossimi allo zero. Questo fenomeno è detto *cold start* e si verifica a causa della mancanza di dati precedenti all'inizio della serie temporale. In particolare, la forte stagionalità giornaliera dei dati (s = 24) e la necessità di differenziazione (che il modello esegue grazie all'ordine D = 1) impedisce al modello di calcolare i primi valori in modo robusto. Il modello ha infatti bisogno di almeno un ciclo giornaliero completo per stabilizzarsi: questo fenomeno è chiaro osservando i primi valori dell'output della funzione `fit_compare_sarimax`, in cui si vedono gli "outliers" al primo step e al ventiquattresimo.

Time	Fitted value
2022-03-01 00:00:00	0.000000
2022-03-01 01:00:00	250.449833
2022-03-01 02:00:00	249.292196
2022-03-01 03:00:00	248.125227
2022-03-01 04:00:00	246.257743
2022-03-01 05:00:00	245.223203
2022-03-01 06:00:00	243.127220
2022-03-01 07:00:00	243.642828
2022-03-01 08:00:00	242.818110
2022-03-01 09:00:00	244.397132
2022-03-01 10:00:00	247.392216
2022-03-01 11:00:00	243.887195
2022-03-01 12:00:00	245.490763

Time	Fitted value
2022-03-01 13:00:00	246.684006
2022-03-01 14:00:00	246.689079
2022-03-01 15:00:00	246.342751
2022-03-01 16:00:00	251.608372
2022-03-01 17:00:00	250.701647
2022-03-01 18:00:00	250.985735
2022-03-01 19:00:00	249.924366
2022-03-01 20:00:00	248.101593
2022-03-01 21:00:00	244.002140
2022-03-01 22:00:00	243.952935
2022-03-01 23:00:00	242.808307
2022-03-02 00:00:00	368.925638
2022-03-02 01:00:00	242.730689

Tabella 3.6: Primi valori dati in output dalla funzione `fit_compare_sarimax`.

Poiché in figura 3.4 non è apprezzabile la differenza tra la previsione e i valori di test, si riporta un ingrandimento della figura, ottenuto escludendo i primi valori e di seguito la tabella delle metriche di errore ottenute:

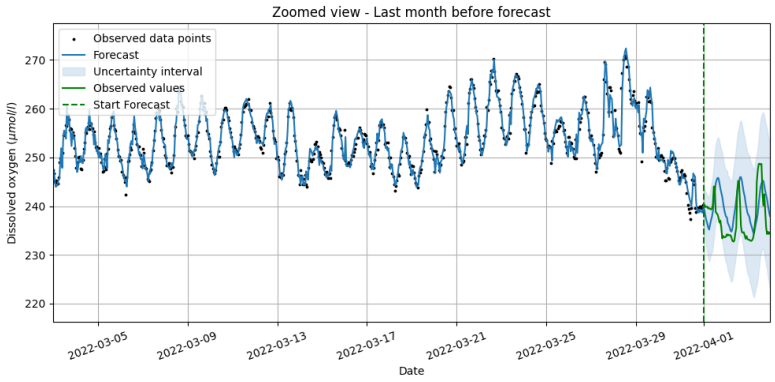


Figura 3.5: Zoom del confronto tra valori reali e previsti dal modello SARIMA univariato.

Metric	**Train**	**Test**	**Total**
MAE	1.742	4.445	1.980
MAPE (%)	0.692	1.874	0.796
RMSSE	5.371	2.678	5.190
mRMSSE	5.371	2.534	5.166

Tabella 3.7: Metriche di valutazione su periodo di train (marzo 2022), test e totale per il modello SARIMA univariato.

La metrica CRPS, calcolata solo per il periodo di test, è pari a 3.001.
Il plot mostra un adattamento ai dati estremamente accurato nella fase di training, tanto da riprodurre quasi perfettamente l'andamento dei dati storici, un risultato atteso per questo tipo di modelli [1]. Anche per la fase di test i risultati sono molto buoni: il modello riesce bene a catturare l'andamento stagionale della serie dell'ossigeno disciolto e a riprodurlo. Questo è confermato dai valori degli errori presi in considerazione: ad esempio, un MAPE minore del 2% è un ottimo risultato.
È necessario specificare che nel calcolo degli errori di train, sono presenti anche i contributi dei primi 24 step, che possono distorcere l'effettiva capacità del modello. In particolare, considerando che la concentrazione media dell'ossigeno all'inizio del periodo di osservazione (mese di marzo) è di circa pari a 246 $\mu mol/l$, il contributo di cui tener conto è pari a 246 $\mu mol/l$ $\cdot$ 2 step $=$ 492 $\mu mol/l$. Se tale valore viene "spalmato" sull'intero periodo di train, composto da 744 osservazioni, l'apporto finale del *cold start* sull'errore è di 492 $\mu mol/l$: 744 step $= 0.66$ $\mu mol/l$. Per ottenere dunque un risultato più significativo, è necessario sottrarre questo valore alla metrica di train. Tuttavia, poiché le valutazioni più significative quelle relative al periodo di test, l'effetto del cold start sul train può essere considerato trascurabile ai fini dell'analisi complessiva.

Per avere un confronto, si procede applicando la funzione implementata all'intero periodo a disposizione, eccetto gli ultimi 3 giorni di osservazioni, che verranno utilizati come test set. Si ottiene la figura seguente:

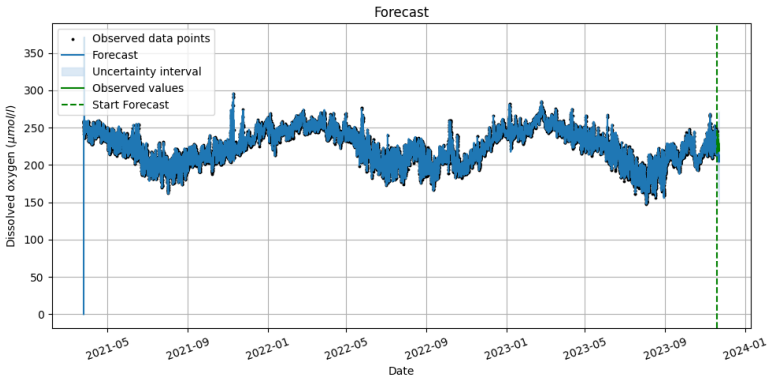


Figura 3.6: Confronto tra valori reali e previsti dal moello SARIMA univariato.

Ancora una volta, i valori iniziali sono rappresentativi del fenomeno di *cold start*, come mostrato dai primi valori della serie restituiti in output da `fit_compare_sarimax` di seguito:

Time	Fitted value
2021-03-25 11:00:00	0.000000
2021-03-25 12:00:00	0.027536
2021-03-25 13:00:00	0.053137
2021-03-25 14:00:00	0.076631
2021-03-25 15:00:00	0.098223
2021-03-25 16:00:00	0.117975
2021-03-25 17:00:00	0.135793
2021-03-25 18:00:00	0.151708
2021-03-25 19:00:00	0.165757
2021-03-25 20:00:00	0.177890
2021-03-25 21:00:00	0.188285
2021-03-25 22:00:00	0.197131
2021-03-25 23:00:00	0.204344

Time	Fitted value
2021-03-26 00:00:00	0.209789
2021-03-26 01:00:00	0.213606
2021-03-26 02:00:00	0.215774
2021-03-26 03:00:00	0.216481
2021-03-26 04:00:00	0.215649
2021-03-26 05:00:00	0.213428
2021-03-26 06:00:00	0.209674
2021-03-26 07:00:00	0.204543
2021-03-26 08:00:00	0.198104
2021-03-26 09:00:00	0.190339
2021-03-26 10:00:00	0.181388
2021-03-26 11:00:00	256.736798
2021-03-26 12:00:00	247.258921

Tabella 3.8: Primi valori dati in output dalla funzione `fit_compare_sarimax`.

In questo caso, i valori dei primi 23 step sono tutti fuori range: la motivazione risiede nella lunghezza del dataset di train. Con un mese di dati, il modello ha accesso a pochi cicli stagionali (circa 30), quindi la parte stagionale è approssimata rapidamente. Con più di due anni di dati invece, il modello ha centinaia di cicli da stimare: le matrici interne usate nel processo di ottimizzazione partono da valori più complessi e richiedono più passi per arrivare a stabilizzazione.

Per osservare la previsione, limitandosi agli ultimi giorni dell'output, si riporta un ingrandimento della figura 3.6:

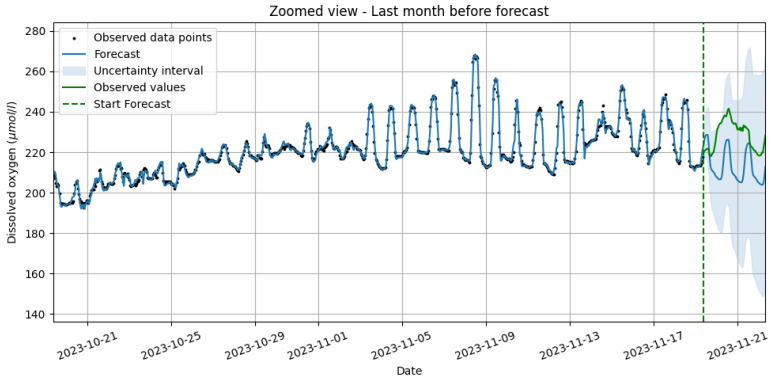


Figura 3.7: Zoom del confronto tra valori reali e previsti dal modello SARIMA univariato.

Anche in questo caso, il modello è molto accurato nell'adattamento ai dati. Tuttavia, nella fase di test, le previsioni non risultano altrettanto precise: il modello non riesce a generalizzare completamente, bensì tende a sottostimare i valori reali, nonostante il lungo periodo dato a disposizione come train set. È da considerare però, che i giorni scelti come test presentano alcune variazioni strutturali, forse dovute a fenomeni metereologici improvvisi. Per questo motivo il modello, basandosi esclusivamente sulla componente autoregressiva e su quella a media mobile, non riesce a catturare tali variaizoni in modo del tutto efficace.
Si nota inoltre un forte aumento dell'incertezza associata alle previsioni: le bande di confidenza risultano infatti molto più ampie rispetto al caso in cui il modello sia addestrato sul solo mese di marzo 2022, poiché la maggiore eterogeneità stagionale e la variabilità complessiva del periodo rendono più difficile il calcolo di stima stabile dei parametri.
Si riportano le metriche relative alla fase di train, a quella di test e a quella totale, che confermano quanto detto finora a proposito della differenza tra bontà del modello nella fase di train e in quella di test, almeno per questo specifico caso:

Metric	**Train**	**Test**	**Total**
MAE	1.904	15.943	1.947
MAPE (%)	0.865	6.932	0.883
RMSSE	1.014	5.161	1.052
mRMSSE	1.014	11.960	1.054

Tabella 3.9: Metriche di valutazione su periodo di train, test e totale (intero periodo disponibile) per il modello SARIMA univariato.

La metrica CRPS, calcolata solo per il periodo di test, è pari a 10.629.

3.5.3 Introduzione dei regressori

L'introduzione di regressori, sotto forma di dataframe aggiuntivo da fornire al parametro `exog` di `SARIMAX`, permette al modello di integrare informazioni esterne per riuscire a prevedere meglio la concentrazione dell'ossigeno disciolto.
Per questa trattazione, è stato deciso di fornire tutte le altre variabili provenienti dal sensore del nodo "Baia" e dalla stazione metereologica di ENEA.
Poiché il modello SARIMA richiede che i regressori siano privi di interruzioni temporali, è stato eseguito un controllo di completezza su ciascuna variabile. Le variabili con buchi consecutivi di dati non superiori a 24 ore sono state interpolate, mentre quelle caratterizzate da periodi di assenza di dati più lunghi sono state escluse dall'analisi. In particolare, questo è risultato necessario per le misure della profondità e del voltaggio della batteria dei sensori, di cui si riporta il grafico con la serie dei valori disponibili,in blu, e gli intervalli di dati mancanti, evidenziati in grigio.

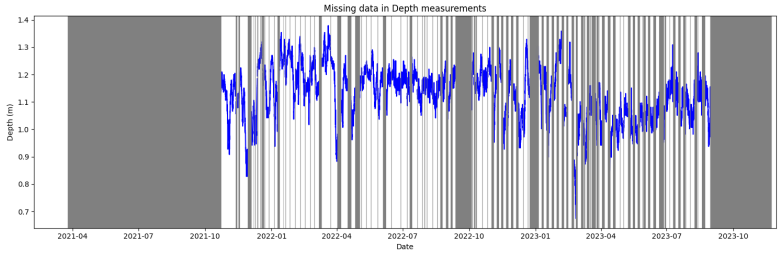


Figura 3.8: Valori mancanti relativi alla Profondità dei sensori.

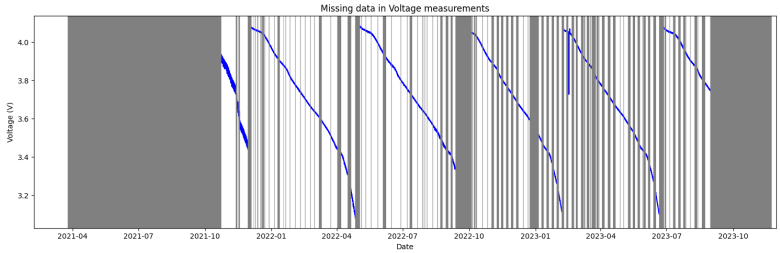


Figura 3.9: Valori mancanti relativi al Voltaggio delle batterie dei sensori.

Applicando la funzione `fit_compare_sarimax` con l'aggiunta di regressori, forniti sotto forma di dataframe, su un mese di train (marzo 2022) e richiedendo la previsione dei 3 giorni successivi, si ottiene:

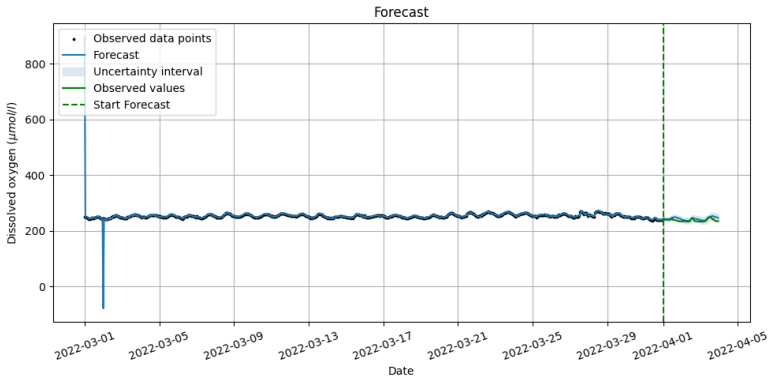


Figura 3.10: Confronto tra valori reali e previsti dal modello SARIMA multivariato.

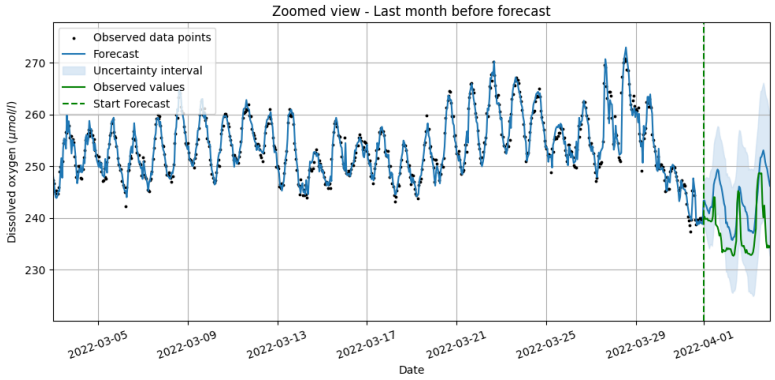


Figura 3.11: Zoom del confronto tra valori reali e previsti dal modello SARIMA multivariato.

Dai plot non sembra esserci un miglioramento rispetto al caso univariato, per cui è necessario un confronto tra le metriche di errore:

Metric	**Train**	**Test**	**Total**
MAE	2.509	5.854	2.805
MAPE (%)	1.001	2.475	1.131
RMSSE	13.672	3.625	13.099
mRMSSE	13.672	3.429	13.040

Tabella 3.10: Metriche di valutazione su periodo di train (marzo 2022), test e totale per il modello SARIMA multivariato.

Il valore del CRPS per la valutazione nel test è pari a 3.001.
Contrariamente a ciò che ci si potrebbe aspettare, confrontando tali metriche con quelle riportate nella tabella 3.7, si nota che il modello multivariato, in questo caso, performa

peggio di quello univariato. La motivazione potrebbe risiedere nella brevità del periodo di addestramento, che riduce la capacità del modello di stimare in modo stabile le relazioni con i regressori.
Un ulteriore aspetto degno di nota riguarda l'andamento del RMSSE e del mRMSSE, che si discosta da quello delle altre metriche: invece di risultare più bassi sul train e più elevati sul test, mostrano il comportamento opposto. Questo fenomeno è causato dal contributo del cold start, che nei primi 24 step produce errori significativamente maggiori. Tali errori incrementano il numeratore del calcolo dell'RMSSE della fase di train, che è dato dalla differenza quadrata media tra i valori fittati dal modello e quelli realmente osservati. Il denominatore, invece, è basato sulla variabilità della serie di train, che resta costante: si ottiene quindi un valore di RMSSE apparentemente più alto sul train, anche se le altre metriche indicano, ragionevolmente, prestazioni migliori. Questa dinamica non si verifica invece nel caso in cui si utilizzi l'intero periodo per le valutazioni del modello, poiché la divisione per il numero di campioni limita tale contributo, come si è osservato nel caso univariato nella tabella 3.7 e come si vedrà in seguito.

Si procede con l'analisi del modello multivariato applicandolo all'intero dataset disponibile. Si ottiene il grafico seguente, limitato all'ultimo mese di osservazione:

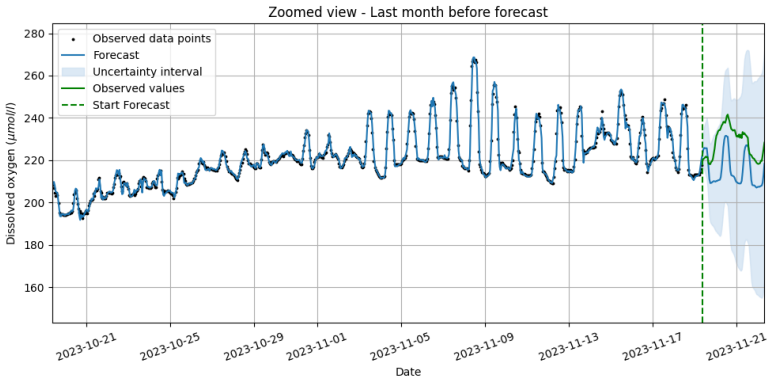


Figura 3.12: Zoom del confronto tra valori reali e previsti dal modello SARIMA multivariato.

Ciò che si può osservare, è che la curva di previsione del modello (in blu) si alza leggermente, avvicinandosi di più ai valori veri rispetto al caso univariato mostrato in figura 3.7. Si ottiene la conferma osservando anche il miglioramento delle metriche ottenute:

Metric	**Train**	**Test**	**Total**
MAE	1.854	13.148	1.889
MAPE (%)	0.842	5.725	0.857
RMSSE	1.156	4.309	1.179
mRMSSE	1.156	9.985	1.181

Tabella 3.11: Metriche di valutazione su periodo di train, test e totale (intero periodo disponibile) per il modello SARIMA multivariato.

Rispetto agli errori ottenuti con gli stessi dati di train e di test, ma senza l'utilizzo di regressori, riportati in tabella 3.9, si può apprezzare una diminuizione di ogni metrica. Ad esempio, il MAE, passando da 15.9 $\mu mol/l$ a 13.1 $\mu mol/l$, diminuisce di quasi 3 unità. Pur trattandosi di valori relativamente alti, si può concludere che l'ausilio delle variabili esogene ha apportato un miglioramento al modello.

Capitolo 4

Il Modello Prophet

Il modello Prophet, sviluppato da Facebook (oggi Meta), è un algoritmo di previsione delle serie temporali concepito per essere al tempo stesso flessibile, interpretabile e adatto anche a non esperti di statistica [16]. È particolarmente efficace in presenza di serie temporali con forti componenti stagionali, che dispongano di abbondanti dati storici.
La libreria è disponibile in Python, oltre che in R, e può essere chiamata con il comando `Prophet()`, personalizzabile grazie a una elevata quantità di parametri da impostare.
Prophet si basa sull'assunzione che la serie temporale $y(t)$ abbia il tempo come unico regressore e che sia scomponibile in modo additivo tramite tre componenti principali e una componente di errore:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t,$$

dove:

1. $g(t)$ è la componente di trend, ovvero la tendenza che ci si aspetta a lungo termine;
2. $s(t)$ è la componente stagionale, ovvero la rappresentazione dei cambiamenti periodici, che possono essere annuali, settimanali o giornalieri;
3. $h(t)$ è l'effetto delle cosiddette *holidays*, che rappresentano degli eventi speciali. Siccome il modello Prophet è stato concepito inizialmente per gestire serie temporali con background economici e commerciali, gli effetti delle holidays erano stati pensati proprio come quei contributi legati al fatto che uno specifico giorno poteva presentare dei dati 'anomali', in quanto giorno di vacanza. Basti pensare ad esempio ad una serie che conta il numero di vendite di un certo negozio di alimentari sfusi. Le vendite di lenticchie, ad esempio, presenteranno un forte aumento nei giorni prima di Capodanno, per poi tornare normali: questo termine viene spiegato dalla componente $h(t)$.
4. ϵ_t è il termine di errore, che si assume essere rumore bianco.

Ogni componente è quindi modellata separamente e in maniera non lineare, ma con un effetto additivo complessivo sulla previsione.

4.1 Il modello di Trend

Prophet prevede due diversi tipi di modelli di trend: un trend lineare a tratti (l'impostazione di default) e un trend di tipo logistico, che saranno esaminati nel dettaglio nelle sezioni seguenti. La tipologia di trend che si vuole adottare è specificabile tramite il parametro `growth = 'linear' / 'logistic'`.

4.1.1 Trend lineare

Il trend lineare è ideale per problemi di forecast con possibilità di crescita illimitata. La formulazione base di questo trend è la seguente:

$$g(t) = kt + m,$$

dove:

- k è il tasso di crescita;
- m è il parametro di offset, cioè l'intercetta iniziale.

Si suppone però che il trend possa cambiare pendenza durante il periodo osservato: questi cambiamenti sono chiamati *changepoints*.
La componente di trend quindi non è più una retta, ma risulta lineare a tratti ed è così definita:

$$g(t) = (k + \mathbf{a}(t)^T\boldsymbol{\delta})t + (m + \mathbf{a}(t)^T\boldsymbol{\gamma}),$$

dove:

- $\boldsymbol{\delta} \in \mathbb{R}^S$ è un vettore che contiene il cambiamento di pendenza δ_j negli S changepoints presenti ai tempi t_j con $j \in \{1, ..., S\}$;
- $\mathbf{a}(t) \in \{0,1\}^S$ è una funzione indicatrice definita come $a_j(t) = \begin{cases} 1, & \text{se } t \geq t_j \\ 0, & \text{altrimenti} \end{cases}$;
- $\boldsymbol{\gamma} \in \mathbb{R}^S$ è un vettore che permette di eseguire degli aggiustamenti al fine di rendere continua la funzione di trend. In particolare è necessario che le sue componenti siano pari a $\gamma_j = -s_j\delta_j$.

La pendenza al tempo t si può quindi esprimere come il tasso di crescita base k più tutti gli aggiustamenti fino a quel punto: $k + \sum_{j:\ t>t_j} \delta_j \ = \ k + \mathbf{a}(t)^T\boldsymbol{\delta}$.

4.1.2 Trend logistico

Adatto per problemi di forecast in cui la misura d'interesse è destinata a crescere in modo non lineare fino ad arrivare a saturazione ad una soglia massima, il trend logistico nella sua forma più basilare si definisce come segue:

$$g(t) = \frac{C}{1 + e^{-k(t-m)}},$$

dove:

- C è la *carrying capacity*, il valore soglia, ed è impostabile aggiungendo una colonna `cap` nel dataframe dei dati di partenza con il valore desiderato. È anche possibile impostare una soglia minima aggiungendo una colonna `floor`.
- k è il tasso di crescita;
- m è il parametro di offset, cioè l'intercetta iniziale.

Questa formulazione però non tiene conto di due importanti aspetti. In primo luogo la carrying capacity C potrebbe non essere costante, ma essere dipendente dal tempo: si sostituisce dunque con $C(t)$. La stessa cosa vale per il tasso di crescita k, che può essere sostituito dalla quantità $k + \mathbf{a}(t)^T\boldsymbol{\delta}$, derivata come nel caso di trend lineare.
La formulazione della componente di trend diventa dunque:

$$g(t) = \frac{C(t)}{1 + e^{-(k+\mathbf{a}(t)^T\boldsymbol{\delta})\ (t-(m+\mathbf{a}(t)^T\boldsymbol{\gamma}))}},$$

dove i vettori $\mathbf{a}(t)$ e $\boldsymbol{\delta}$ sono definiti come nel caso lineare, e $\boldsymbol{\gamma}$ ha lo stesso fine di rendere la funzione $g(t)$ continua, ma in questo caso le sue componenti sono pari a $\gamma_j = \left(t_j - m - \sum_{i<j}\gamma_i\right)\left(1 - \frac{k+\sum_{i<j}\delta_i}{k+\sum_{i\leq j}\delta_i}\right)$.

4.1.3 Changepoints

I changepoints, come anticipato, si riferiscono ai punti della serie temporale in cui le proprietà statistiche della serie quali media, varianza o autocorrelazione cambiano in modo significativo. Possono essere specificati manualmente dall'utente fornendo una lista di date tramite il comando `changepoints = [...]` se è presente una conoscenza pregressa, oppure possono essere individuati automaticamente da Prophet.
Nel primo caso Prophet può comunque evitare di utilizzare i changepoints suggeriti se la media stimata di quello specifico elemento di $\boldsymbol{\delta}$ è prossima a zero.
Nel secondo caso, Prophet dispone di default 25 changepoints equispaziati utilizzando una priori sparsa (che quindi spinge molti coefficienti verso il valore nullo) su $\boldsymbol{\delta}$, come ad esempio $\delta_j \sim \text{Laplace}(0, \tau)$. Questa distribuzione è infatti centrata in zero e la sua "larghezza", e quindi la flessibilità del modello a cambiare la pendenza, è controllata dal parametro $\tau > 0$. Tali changepoints vengono automaticamente inseriti nel primo 80% della serie temporale, per evitare overfitting nelle fluttuazioni finali della serie e quindi per non pesare troppo sulla previsione. È comunque possibile modificare questo parametro di default tramite il comando `changepoint_range = 0.9`, ad esempio.
Un altro strumento di controllo per l'utente è la flessibilità del modello ai cambiamenti di trend: di default si ha che `changepoint_prior_scale = 0.05`. Tale parametro agisce come un moltiplicatore per la priori di Laplace: un valore maggiore (ad esempio 0.5) risulterà in una priori più ampia, con maggiore varianza, permettendo al modello di acquisire molta flessibilità e quindi di considerare changepoints più frequenti. Viceversa, valori bassi del parametro (ad esempio 0.005) rendono la priori più stretta, con varianza minore, e di conseguenza si otterrà un modello più approssimativo che ignorerà piccoli cambiamenti di pendenza e quindi un minor numero di changepoints. In base a tale parametro, Prophet decide il numero di changepoints da tenere in considerazione.

4.2 Il modello di Stagionalità

Prophet permette di catturare gli effetti periodici delle serie temporali tramite la modellizzazione con serie di Fourier standard.
In particolare, supponendo di avere dati storici da modellizzare con stagionalità di periodo P (ad esempio $P = 7$ per dati con pattern settimanali), l'approssimazione degli effetti stagionali che Prophet esegue è data dalla seguente sommatoria di ordine N:

$$s(t) = \sum_{n=1}^{N} \left(a_n \, cos\left(\frac{2\pi nt}{P}\right) + b_n \, sin\left(\frac{2\pi nt}{P}\right) \right) = X(t)\boldsymbol{\beta},$$

dove:

- $X(t) = \left[cos\left(\frac{2\pi(1)t}{P}\right), \; sin\left(\frac{2\pi(1)t}{P}\right), ..., \; cos\left(\frac{2\pi(N)t}{P}\right), \; sin\left(\frac{2\pi(N)t}{P}\right)\right]$ è il vettore "stagionale" relativo al dato del tempo t, contenente i termini armonici;
- $\boldsymbol{\beta} = [a_1, b_1, ..., a_N, b_N]^T$ è il vettore dei $2N$ coefficienti della scomposizione di Fourier. Non hanno un'interpretazione immediata, ma rappresentano il peso per il quale viene moltiplicato il corrisponente fattore sinusoidale.
  L'assunzione per il modello in questo caso è una priori smooth: $\boldsymbol{\beta} \sim \text{Normal}(0, \sigma^2)$.

Le stagionalità annuali e settimanali sono incluse di default se la serie contiene almeno due cicli completi di ciascun tipo, mentre quella giornaliera viene aggiunta in caso di dati a cadenza oraria. Tutte e tre possono essere attivate o disattivate ed è anche possibile aggiungere delle stagionalità personalizzate tramite il metodo `add_seasonality`, che richiede in input un nome identificativo per la stagionalità, il periodo caratteristico misurato in giorni e l'ordine di Fourier.
Una volta attivate, le componenti stagionali sono modellizzate quindi con la serie di Fourier troncata, la cui complessità dipende dal numero N di armoniche della serie. Questo parametro agisce come una sorta di filtro passa-basso: maggiore è N, maggiore sarà la flessibilità e il dettaglio della stagionalità della serie, ma anche il rischio di overfitting. Viceversa, diminuendo N l'approssimazione risulterà più liscia e meno ondulata.
Come impostazione predefinita si ha $N = 10$ per la stagionalità annuale e giornaliera e $N = 3$ per quella settimanale. Questi valori possono essere modificati tramite il parametro `yearly_seasonality = N`, e analogamente per `weekly_seasonality` e `daily_seasonality`.

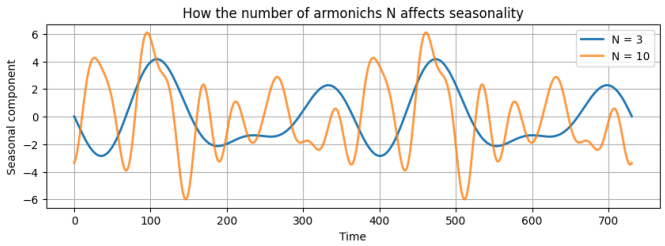


Figura 4.1: Curva di stagionalità in base al numero di armoniche N.

4.3 Le Holidays e i regressori

La componente delle holidays permette di notificare al modello che in quelle specifiche date è successo o succederà qualcosa di anomalo, degli eventi speciali che non possono essere spiegati solo dagli altri termini, in quanto non seguono un pattern periodico.
Per includere le holidays nel modello, è necessario creare un dataframe con due colonne: `holidays` con il nome dell'evento e `ds` per la data in cui si verifica (può essere sia passata che futura, ma deve essere una data compresa nel set di date che comprende train e test set). Inoltre, aggiungendo altre due colonne `lower_window` e `upper_window` è possibile estendere la durata di quella holiday per più giorni, prima (inserendo un numero negativo) e dopo (inserendo un numero positivo) la data specificata in `ds`.
Una volta creato il dataframe, si passa al modello tramite l'argomento `holidays = df`.
Prophet permette anche di includere dei dataframe di holidays già costruiti e specifici per un certo paese, che contiene le relative festività e vacanze nazionali. Questa possibilità risulta utile per serie temporali di argomento economico, commerciale e sociale ed è implementabile con il metodo `add_country_holidays(country_name = 'US')`, ad esempio.
Si assume infine che gli effetti delle holidays siano indipendenti tra loro.
Per ogni holiday $i \in \{1, ..., L\}$, sia D_i il set di date per quella holiday e κ_i il suo effetto, che viene considerato nella previsione. È possibile rappresentare la componente di holidays della serie utilizzando un vettore di funzioni indicatrici $Z(t)$:

$$h(t) = Z(t)\boldsymbol{\kappa} = [\mathbb{1}_{\{t \in D_1\}}, \ldots, \mathbb{1}_{\{t \in D_L\}}]\boldsymbol{\kappa}.$$

Allo stesso modo rispetto a quanto detto per la stagionalità, come priori si utilizza $\boldsymbol{\kappa} \sim \text{Normal}(0, \nu^2)$.

Un altro modo di fornire informazioni al modello è quello di aggiungere dei regressori tramite il metodo `add_regressor`. Il vantaggio, a differenza delle holidays, è che permette di inserire dati di tipo anche non binario. Per esempio, è possibile usare un'altra serie temporale come regressore, con il vincolo che i suoi valori relativi al periodo di previsione siano noti (oppure predetti separatamente, mettendo in conto che l'errore nella prima previsione si propagherà nella seconda) e che non siano costanti nell'intera durata.
Il contributo dei regressori $r(t)$, al pari di quello delle holidays, si aggiunge in modo additivo alla previsione $y(t)$ e può essere visto come segue:

$$r(t) = R(t)\boldsymbol{\rho},$$

dove:

- $R(t) = [\boldsymbol{r_1}(t), ..., \boldsymbol{r_K}(t)]$ è la matrice dei K regressori, che deve essere fornita dall'utente;
- $\boldsymbol{\rho} = [\rho_1, ..., \rho_K]^T$ è il vettore dei coefficienti regressivi, che rappresentano l'incremento della predizione per un incremento unitario del valore del regressore. L'assunzione per la priori è $\boldsymbol{\rho} \sim \text{Normal}(0, \sigma^2)$.
  Per ottenere la stima di Prophet dei loro valori esiste la funzione `regressor_coefficients`.

4.4 Stime dei parametri e dell'incertezza

Prophet utilizza la stima del massimo a posteriori (MAP) per la valutazione dei parametri del modello. Tale metodo sfrutta la massimizzazione della distribuzione a posteriori, che è la probabilità del parametro date le osservazioni.
In particolare, siano $\boldsymbol{\theta}$ i parametri e $\mathbf{x}$ i dati osservati, sia f la relativa distribuzione campionaria, tale che $f(\mathbf{x}|\boldsymbol{\theta})$ sia la probabilità di $\mathbf{x}$ dato il parametro $\boldsymbol{\theta}$. Se si suppone inoltre che il parametro $\boldsymbol{\theta}$ sia la realizzazione di una variabile aleatoria $\boldsymbol{\Theta}$ e abbia una distribuzione a priori $g_{\boldsymbol{\Theta}}$ derivata da una conoscenza pregressa, è possibile sfruttare il teorema di Bayes e ottenere la distribuzione a posteriori di $\boldsymbol{\theta}$:

$$f(\boldsymbol{\theta}|\mathbf{x}) = \frac{f(\mathbf{x}|\boldsymbol{\theta})g_{\boldsymbol{\Theta}}(\boldsymbol{\theta})}{\int_{\boldsymbol{\theta}} f(\mathbf{x}|\boldsymbol{\theta})g_{\boldsymbol{\Theta}}(\boldsymbol{\theta})}$$

A questo punto, la stima puntuale del parametro si ottiene con la massimizzazione di tale posteriori:

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} f(\boldsymbol{\theta}|\mathbf{x}) = \arg\max_{\boldsymbol{\theta}} \frac{f(\mathbf{x}|\boldsymbol{\theta})g_{\boldsymbol{\Theta}}(\boldsymbol{\theta})}{\int_{\boldsymbol{\theta}} f(\mathbf{x}|\boldsymbol{\theta})g_{\boldsymbol{\Theta}}(\boldsymbol{\theta})} = \arg\max_{\boldsymbol{\theta}} f(\mathbf{x}|\boldsymbol{\theta})g_{\boldsymbol{\Theta}}(\boldsymbol{\theta})$$

dove il secondo passaggio è giustificato dal fatto che il denominatore non dipende dal valore di $\boldsymbol{\theta}$ e quindi può essere ignorato.
Questa tecnica si differenzia dalla stima di massima di verosimiglianza (ML) perché assume l'esistenza di una conoscenza pregressa sul parametro, infatti la stima ML si basa solo sulla massimizzazione della verosimiglianza, ovvero la probabilità dei dati dato il parametro:

$$\hat{\boldsymbol{\theta}}_{\text{ML}} = \arg\max_{\boldsymbol{\theta}} f(\mathbf{x}|\boldsymbol{\theta})$$

Al contrario, le stime MAP integrano l'informazione proveniente dalla distribuzione a priori $g_{\boldsymbol{\Theta}}$.
Tuttavia, trattandosi di stime puntuali, non restituiscono la distribuzione completa dei parametri né le relative incertezze, e questo rappresenta un limite significativo rispetto all'approccio bayesiano completo.

Relativamente all'incertezza, Prophet adotta un'impostazione semplificata: di default considera come unica fonte di variabilità i possibili cambi futuri di trend, e per questo motivo restituisce solo gli intervalli associati a questa componente. In particolare, assume che il modello del trend stimato in fase di training rimanga valido anche nel futuro: i cambiamenti di rate sono quindi ancora distribuiti secondo $\delta_j \sim \text{Laplace}(0, \tau)$, sostituendo τ con una stima della varianza λ calcolata in base ai dati. Maggiore risulterà τ, maggiore sarà la flessibilità ai cambiamenti di rate.
I changepoints futuri sono campionati in modo tale che la loro frequenza sia simile a quella dei T dati storici, in cui erano presenti S changepoints. In particolare $\forall j > T$,

$$\begin{cases} \delta_j = 0 & \text{con probabilità } \frac{T-S}{T} \\ \delta_j \sim \text{Laplace}(0, \lambda) & \text{con probabilità } \frac{S}{T} \end{cases}$$

L'assunzione che il trend futuro segua lo stesso andamento di quello storico è però una forte supposizione: di conseguenza, gli intervalli di incertezza potrebbero non coprire completamente i valori reali. Aumentando τ, il modello diventa più flessibile nel periodo di training, e quindi l'errore corrispondente diminuisce; ma la stessa flessibilità, proiettata nel periodo di test, si traduce in intervalli di incertezza più ampi.
Di default la copertura degli intervalli è fissata all'80%, ma questo valore è modificabile tramite il comando `interval_width = 0.95`, ad esempio.

Per ottenere anche gli intervalli di incertezza della previsione relativa agli altri parametri quali stagionalità, holidays e regressori, è necessario che Prophet esegua un campionamento bayesiano completo. Questa modalità si attiva impostando `mcmc_samples` ad un valore maggiore di zero (è consigliato usarne almeno 300), corrispondente al numero di campioni MCMC (Markov Chain Monte Carlo) da generare per ogni parametro. Ciò consente di ottenere le posteriori dei coefficienti della serie di Fourier e di conseguenza le loro incertezze, a fronte però di costi computazionali molto più elevati.
In particolare, la tecnica utilizzata da Prophet per le MCMC è il No-U-Turn Sampler (NUTS), un algoritmo efficiente ed auto-adattativo, ovvero che non richiede un tuning manuale, ed è particolarmente adatto per modelli complessi come Prophet, dove il numero di parametri può essere elevato.

4.5 Applicazione del modello Prophet

In questa sezione vengono presentati e discussi i risultati ottenuti applicando il modello Prophet alla serie temporale in esame.
L'obiettivo è quello di capire il funzionamento pratico del moello e valutarne le performance, confrontando le sue previsioni con i dati reali ed evidenziando quale sia l'influenza dei parametri impostati manualmente.
Innanzitutto è stato necessario creare una funzione ad hoc per facilitare tale confronto:

```
def fit_compare(df_tot, start_train, start_forecast, end_forecast,
    growth = 'linear', changepoints = None, show_changepoints = False,
    changepoint_prior_scale = 0.05, changepoint_range = 0.8, holidays =
    None, regressors = None, mcmc_samples = 0, plot = True,
    show_components = True, title = 'Results'):
```

Listing 4.1: fit_compare function

Tale funzione riceve il dataset completo sotto forma di Pandas dataframe: in particolare Prophet richiede che il dataset abbia una colonna chiamata `ds` (datestamp) formata dalle date, e una seconda colonna chiamata `y`, di tipo numerico, contenente le misurazioni del parametro da osservare: il dataframe non rappresenta nient'altro che una serie temporale. La funzione suddivide quindi i dati in training set e test set sulla base delle date specificate in input (`start_train, start_forecast, end_forecast`). In seguito, viene effettuato un controllo sul parametro `growth`, che definisce la tipologia di trend da utilizzare nel modello: nel caso della serie dell'ossigeno, viene impostato di default il trend lineare. Qualora invece si specificasse una crescita di tipo logistico, la funzione aggiungerebbe al

dataframe la colonna `cap`, contenente il valore di capacità di saturazione, come descritto nella sezione 4.1.2. Tuttavia, questa configurazione non è stata adottata nell'analisi in oggetto.
Successivamente, viene creato l'oggetto Prophet, che viene configurato con i parametri relativi alla configurazione dei changepoints: `changepoints` per specificare una lista di date, `show_changepoints` per mostrarli o meno nei plot, `changepoint_prior_scale` per controllare la flessibilità ai cambiamenti di trend, e `changepoint_range` per indicare la frazione del periodo totale in cui il modello può cercare cambiamenti di trend. Altri parametri sono `holidays` e `regressors` per l'aggiunta di eventi speciali e regressori e infine `mcmc_samples` per il campionamento bayesiano opzionale.
Il modello viene quindi addestrato sui dati di training e successivamente usato per generare previsioni sull'intero intervallo temporale, composto sia dai dati di training che da quelli di test.
La funzione restituisce in output il dataframe `forecast` prodotto da Prophet, che include l'intero periodo temporale che comprende le date del training e del test set e che, per ciascuna di esse, riporta:

- la previsione del trend (`trend`) e i relativi intervalli di incertezza (`trend_lower, trend_upper`);
- il contributo delle stagionalità presenti (tra `daily, weekly, yearly`) e i relativi intervalli di incertezza;
- il contributo aggiuntivo delle componenti quali holidays e regressori (`additive_terms`) e la relativa incertezza (`additive_terms_lower, additive_terms_upper`);
- la previsione centrale (`yhat`) e i relativi intervalli di incertezza (`yhat_lower, yhat_upper`), ottenuti dalla somma dei precedenti termini.

È importante far notare che se il campionamento bayesiano opzionale non è stato attivato, gli unici estremi degli intervalli di incertezza che non saranno uguali alla previsione centrale saranno quelli del trend, come visto nella sezione 4.4.
Inoltre, per visualizzare graficamente i risultati, la funzione restituisce tre plot:

1. il primo mostra il confronto tra la curva stimata (colorata di blu) e quella osservata (rappresentata dai pallini neri per il training e da una curva verde nel test);
2. il secondo mostra i contributi di ogni componente del modello lungo tutto il periodo: saranno presenti quindi il plot del trend, il plot delle stagionalità presenti, il plot delle holidays e quello dei regressori;
3. il terzo mostra l'andamento dell'errore assoluto commesso, insieme a due linee tratteggiate che rappresentano l'errore medio (Mean Absolute Error) nel fit del training test (in giallo) e l'errore medio nella previsione nel test set (in rosso).

Per mostrare un'applicazione pratica della funzione `fit_compare` e dei meccanismi di Prophet, verrà utilizzato un periodo di un mese come training set al fine di stimare il modello, e saranno generate previsioni sui tre giorni successivi. In particolare, per questa e per le successive analisi, si sceglie il mese di marzo 2022, quindi si imposta:

- `start_train` = '2022-03-01 00:00:00'
- `start_forecast` = '2022-04-01 00:00:00'
- `end_forecast` = '2022-04-03 23:00:00'

Lasciando le impostazioni di default, senza aggiungere nè la presenza di holidays nè quella di regressori, il risultato che si ottiene è il seguente:

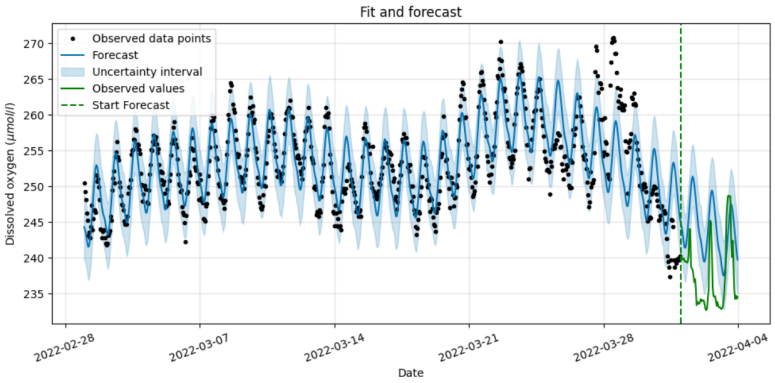


Figura 4.2: Confronto tra valori reali e previsti dal modello Prophet per il mese di marzo 2022.

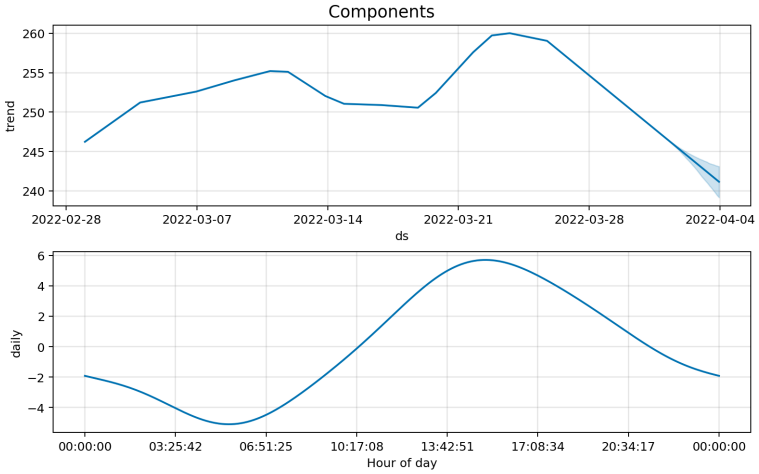


Figura 4.3: Componenti della previsione.

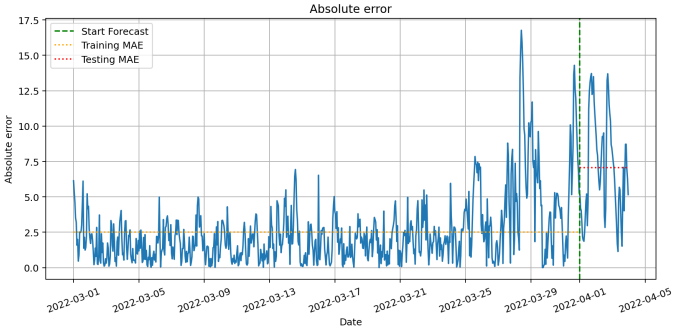


Figura 4.4: Errori assoluti ottenuti.

Come è possibile notare, nonostante Prophet sia un modello relativamente semplice, il fit è abbastanza preciso, così come la previsione. Il massimo errore raggiunto è di circa 16 $\mu mol/l$ nella fase di training, seguito da errori di circa 13 $\mu mol/l$ nella fase di test.
In generale però l'errore medio sull training è minore di quello di test, come evidenziato dalle linee tratteggiate in figura 4.5. Questo comportamento è in parte un risultato atteso poiché il modello riesce ad adattarsi meglio ai dati del training, siccome li ha a disposizione nella fase di apprendimento, mentre ha più difficoltà a generalizzare su dati futuri. Tale scarto abbastanza contenuto tra i due errori significa anche che il modello non è andato incontro ad underfitting, ovvero non è troppo semplicistico.
Tuttavia, un'analisi basata esclusivamente sull'errore assoluto (espresso in $\mu mol/l$) potrebbe fornire un'informazione parziale sulla bontà del modello. Infatti, un errore medio di circa 13 $\mu mol/l$ può avere un significato molto diverso a seconda dell'ordine di grandezza dei valori di ossigeno disciolto osservati: se i livelli medi sono elevati, tale errore rappresenta una piccola deviazione percentuale; se invece i valori sono bassi, lo stesso valore può corrispondere a un errore relativo molto più significativo.
Per questo motivo è utile affiancare all'errore assoluto anche una misura percentuale, come l'APE (Absolute Percentage Error) e la sua versione mediata, il MAPE (Mean Absolute Percentage Error, 2.2.2), che normalizzano la deviazione in funzione del valore osservato. In questo modo, seppur con delle limitazioni in presenza di valori prossimi a zero, problema che non sussiste nella serie dell'ossigeno, è possibile interpretare meglio l'errore, indipendentemente dall'unità di misura e dal range dei dati.
A tal fine, è stata implementata una funzione apposita che calcola anche l'errore percentuale per ciascun punto e la sua media sull'insieme di training e su quello di test, così da confrontare le prestazioni del modello in modo più robusto e informativo.
In questo contesto, anche l'RMSSE (Root Mean Square Scaled Error) e la sua versione modificata (2.2.3) rappresentano alternative utili, perché normalizzano l'errore quadratico medio rispetto alla variabilità intrinseca dei dati, che può penalizzare la previsione. Inoltre, essendo basati sul quadrato degli errori, penalizzano maggiormente le previsioni con grandi scostamenti, risultando sensibili ai picchi e agli outliers.
L'uso combinato di queste metriche consente quindi una valutazione più completa delle prestazioni del modello e i loro valori saranno riportati sotto forma di tabella riassuntiva, con una suddivisone per periodo temporale: fase di training, fase di test e totale.

In questo caso:

Metric	Train	Test	Total
MAE	2.507	7.073	2.910
MAPE (%)	0.988	2.992	1.165
RMSSE	1.811	4.088	2.113
mRMSSE	1.811	3.868	2.103

Tabella 4.1: Metriche di valutazione su periodo di train, test e totale (marzo 2022) per il modello Prophet univariato.

Un altro aspetto interessante emerge dall'analisi delle componenti estratte per il mese di marzo: nella componente giornaliera si osserva che, nelle prime ore del giorno (circa tra le 7 e le 13), la concentrazione di ossigeno disciolto mostra un incremento quasi lineare. Si riporta di seguito il plot relativo a due giorni consecutivi per evidenziare meglio tale andamento:

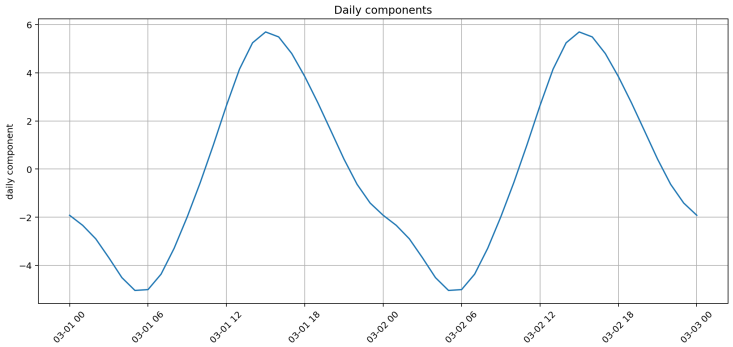


Figura 4.5: Componente giornaliera per due giorni consecutivi di marzo.

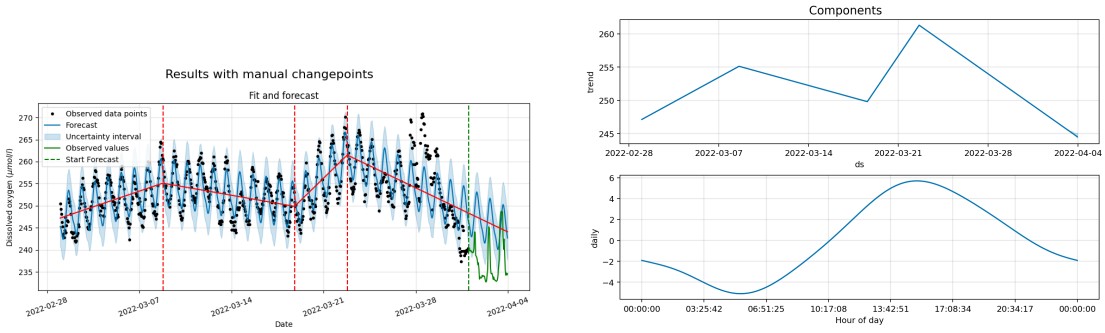
Questo fenomeno può essere attribuito principalmente all'attività fotosintetica del fitoplancton, che inizia ad agire con l'aumento della radiazione solare dopo l'alba. Durante la fotosintesi, infatti, il fitoplancton rilascia ossigeno nell'acqua, contribuendo a un arricchimento progressivo della sua concentrazione nelle ore iniziali della giornata.
Durante le ore notturne invece, si ha un comportamento simile ma inverso: l'ossigeno diminusice in modo graduale a causa della respirazione degli organismi (fitoplancton stesso, zooplancton, batteri ecc). Si noti anche la presenza di un piccolo "scalino" intorno all'1 di notte: tale discontinuità può essere interpretata come il risultato congiunto di processi biologici e fisici, per esempio rimescolamenti dovuti a variazioni di densità, correnti o cambiamenti nelle condizioni meteorologiche locali.
Queste evidenze confermano l'importanza e la complessità dei processi biologici e non solo, che regolano le dinamiche dell'ossigeno disciolto in ambiente costiero.

4.5.1 Adattabilità del trend

Come discusso nella sezione 4.1.3, esistono due modi per personalizzare la capacità del modello di stimare il trend: il primo consiste nel fornire esplicitamente una lista di date (tramite `changepoints = [...]`) nelle quali si ritiene che la serie abbia subito una variazione di pendenza; il secondo prevede la modifica del livello di flessibilità del modello ai cambiamenti di trend (tramite `changepoint_prior_scale`).
Inoltre si specifica che, in questa trattazione, è stato adottato un trend lineare anziché logistico, poiché non sono presenti limiti superiori o inferiori significativi: il livello di saturazione dell'ossigeno, pur potendo rappresentarli teoricamente, non è indicativo a causa delle frequenti condizioni di sovra e sottosaturazione.

Scelta dei changepoints: automatica o manuale

Osservando l'andamento della serie temporale, si individuano tre principali date di "rottura" del trend: 8 marzo, 18 marzo, 22 marzo (indicate dalle rette tratteggiate in rosso). Fornendole al modello sotto forma di lista, si ottiene il seguente risultato:



(a) Confronto tra valori reali e previsti.

(b) Componenti della previsione.

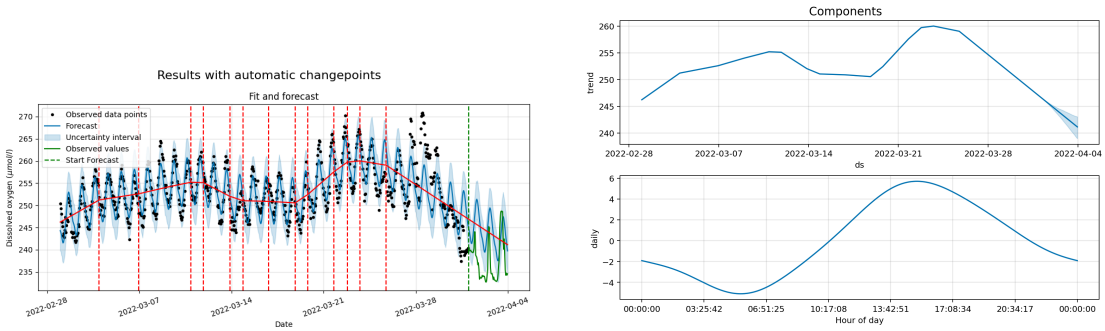
Figura 4.6: Risultati ottenuti con changepoints manuali per marzo 2022.

La previsione tende a sovrastimare i dati effettivi nel periodo di test: questo comportamento si ha perché Prophet rimane ancorato all'ultimo cambiamento di trend indicato dalle date fornite, che è una decina di giorni prima del periodo di previsione, e non attribuisce un peso sufficiente all'andamento più recente della serie.
Con tale configurazione, gli errori ottenuti sono i seguenti:

Metric	**Train**	**Test**	**Total**
MAE	2.584	8.853	3.138
MAPE (%)	1.020	3.748	1.261
RMSSE	1.874	5.025	2.331
mRMSSE	1.874	4.755	2.320

Tabella 4.2: Metriche ottenute con changepoints manuali per marzo 2022.

Lasciando invece a Prophet la libertà di inserire i changepoints in modo automatico, si ottiene questo risultato:



(a) Confronto tra valori reali e previsti.

(b) Componenti della previsione.

Figura 4.7: Risultati ottenuti con changepoints automatici per marzo 2022.

Si nota subito che in questo scenario i changepoints automatici sono più numerosi (13 in totale, più della metà dei 25 inseriti in modo equispaziato da Prophet) e coincidono solo in parte a quelli inseriti manualmente: ad esempio, l'8 marzo non è proprio preso in considerazione, a differenza delle altre due date.
Anche in questo caso la previsione tende a sovrastimare i valori reali, sempre poiché Prophet posiziona di default i changepoints nel primo 80% della serie, attribuendo una rilevanza limitata ai dati più recenti. Questo aspetto, soprattutto nelle analisi che utilizzano un periodo temporale più ampio, risulterà più problematico, e sarà quindi necessario modificare tale parametro (`changepoint_range`).
La discrepanza tra i grafici di quest'ultima previsione e la precedente non è molto visibile a occhio nudo, per cui si riportano nuovamente le tabelle delle metriche per un confronto più immediato:

Metric	**Train**	**Test**	**Total**
MAE	2.507	7.073	2.910
MAPE (%)	0.988	2.992	1.165
RMSSE	1.811	4.088	2.113
mRMSSE	1.811	3.868	2.103

Tabella 4.3: Metriche ottenute con changepoints automatici per marzo 2022.

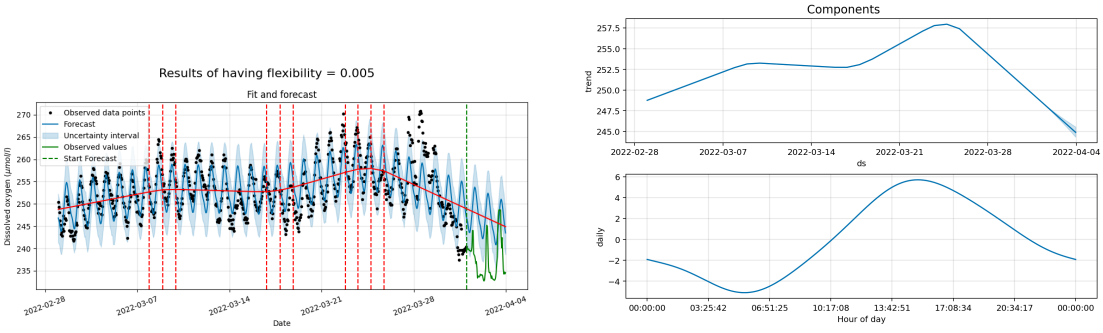
Si evince che in generale le prestazioni del modello con changepoints automatici sono migliori rispetto al modello con i changepoints forzati manualmente. Questo aspetto può essere giustificato dal fatto che i changepoints individuati da Prophet sono 10 in più rispetto a quelli inseriti manualmente. Infatti, confrontando i plot delle componenti, si può notare come nel caso dei changepoints manuali la linea spezzata del trend presenti tre punti angolosi ma molto pronunciati, mentre nel caso dei changepoints automatici la spezzata

appaia più smooth, grazie alla maggiore frequenza di punti angolosi. Ragionevolmente, questo permette maggiore precisione e maggiore adattabilità.

Flessibilità alle variazioni del trend

Il parametro `changepoint_prior_scale` è uno strumento per controllare la flessibilità del modello ai cambiamenti del trend: valori elevati portano a modelli più sensibili mentre valori minori rendono i modelli più rigidi.
Per visualizzare l'effetto di tale parametro sulla previsione e sulle stime del modello, si riportano tre confronti caratterizzati da flessibilità diverse:

1. `changepoint_prior_scale = 0.005`



(a) Confronto tra valori reali e previsti.

(b) Componenti della previsione.

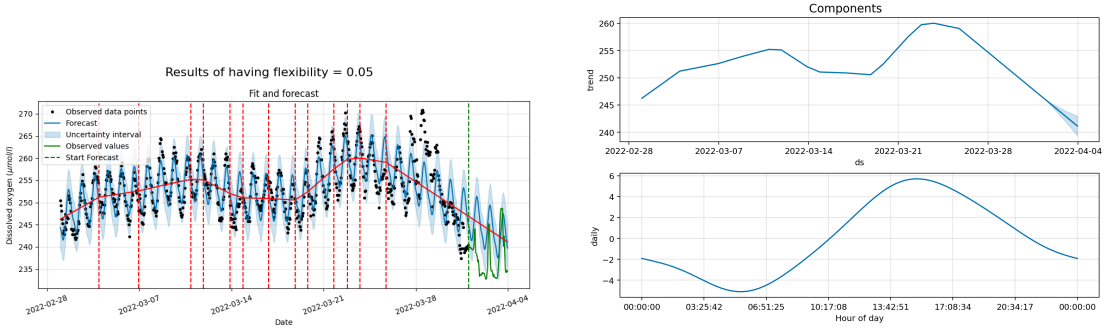
Figura 4.8: Risultati ottenuti con flessibità pari a 0.005 per marzo 2022.

Metric	**Train**	**Test**	**Total**
MAE	2.950	9.457	3.524
MAPE (%)	1.163	4.004	1.414
RMSSE	2.027	5.341	2.503
mRMSSE	2.027	5.053	2.492

Tabella 4.4: Metriche ottenute con flessibità pari a 0.005 per marzo 2022.

Si noti che le metriche di questo modello, nonostante abbia 10 changepoints, sono peggiori di quelle del modello ottenuto nella sezione precedente con l'inserimento manuale di 3 changepoints (tabella 4.2).

2. `changepoint_prior_scale = 0.05` (default, del tutto uguale al caso con changepoints automatici della sezione precedente, ma che viene riportato per comodità del confronto)



(a) Confronto tra valori reali e previsti.

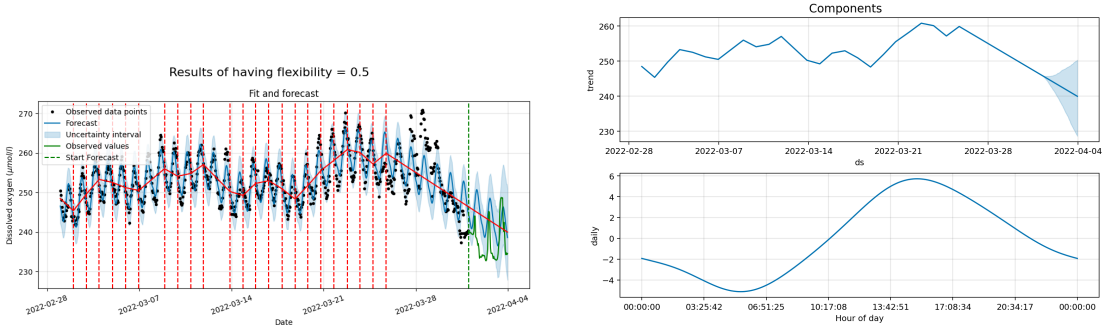
(b) Componenti della previsione.

Figura 4.9: Risultati ottenuti con flessibilità pari a 0.05 per marzo 2022.

Metric	**Train**	**Test**	**Total**
MAE	2.507	7.073	2.910
MAPE (%)	0.988	2.992	1.165
RMSSE	1.811	4.088	2.113
mRMSSE	1.811	3.868	2.103

Tabella 4.5: Metriche ottenute con flessibilità pari a 0.05 per marzo 2022.

3. `changepoint_prior_scale = 0.5`



(a) Confronto tra valori reali e previsti.

(b) Componenti della previsione.

Figura 4.10: Risultati ottenuti con flessibilità pari a 0.5 per marzo 2022.

Metric	Train	Test	Total
MAE	2.180	6.400	2.552
MAPE (%)	0.858	2.705	1.021
RMSSE	1.690	3.759	1.963
mRMSSE	1.690	3.557	1.954

Tabella 4.6: Metriche ottenute con flessibilità pari a 0.5 per marzo 2022.

In questa analisi, incrementando `changepoint_prior_scale` da 0.005 a 0.5 si può osservare innanzitutto l'aumento del numero di changepoints: partendo da 10, per poi averne 13 e infine 23. Questo è risultato in modelli via via più precisi e flessibili, e dunque in una diminuzione delle metriche di errore in generale. Tuttavia, si può notare un sensibile aumento dell'ampiezza delle bande di incertezza nel periodo di test, dove era richiesta la previsione. Ciò è coerente con il classico trade-off bias–variance: la maggiore flessibilità permette un migliore adattamento ai dati storici e alla riduzione dell'errore assoluto, ma aumenta la varianza delle stime della pendenza finale, traducendosi in maggiore incertezza delle previsioni.
Per selezionare il valore ottimale del parametro in questione si è deciso di mantenere il valore più parsimonioso che comporta comunque buoni valori delle metriche per garantire previsioni più robuste, ovvero `changepoint_prior_scale = 0.05`, che è anche il valore di default.
Il terzo caso infatti, seppur corrispondente ai valori minimi degli errori, presenta changepoints distribuiti a cadenza quasi giornaliera, considerando che il periodo temporale considerato è di un mese. Questo fenomeno potrebbe portare a overfitting, perché il modello si adatta troppo al rumore della serie. In pratica, i changepoints sembrano cercare di catturare la stagionalità giornaliera, producendo informazioni ridondanti che non aggiungono valore predittivo.
Tali analisi sono state effettuate anche modificando il parametro `changepoints_range` ed aumentandolo da 0.8 fino a 0.9 e 0.95, ma le conclusioni rimangono le medesime.

4.5.2 Introduzione delle holidays

Per affinare il modello e incorporare l'impatto di eventi significativi che non vengono catturati dalla stagionalità naturale dei dati, si introducono le holidays, ovvero gli eventi "straordinari" trattati nella sezione 4.3. Prophet richiede che le holidays vengano passate tramite un dataframe contenente una colonna `ds` con le date (che possono appartenere al periodo dei dati osservati o essere successive, al fine di migliorare la previsione) e una colonna con il nome dell'evento speciale. Inoltre, aggiungendo altre due colonne `lower_window` e `upper_window` è possibile estendere la durata di quella holiday per più giorni, prima (inserendo un numero negativo) e dopo (inserendo un numero positivo) la data specificata in `ds`.
L'idea è quella di individuare i giorni corrispondenti a fenomeni atmosferici anomali che possano aver contribuito alla modifica dei livelli di ossigeno disciolto nella baia.
È noto, per esempio, che in generale la relazione tra ossigeno disciolto e la salinità sia

inversamente proprozionale: l'acqua salata ha una minore capacità di sciogliere i gas (in particolare l'acqua salata trattiene il 20% in meno di ossigeno disciolto) poiché le molecole di sale occupano spazio e interagiscono con quelle d'acqua. Inoltre, relativamente vicino alla baia sono presenti le foci di due fiumi: il Magra e l'Arno, che contribuiscono ad immettere acqua dolce nella zona interessata, e quindi ad abbassarne la salinità. Per questo motivo è stato deciso di inserire tra le holidays i giorni con livello del fiume particolarmente alto.
Un altro fattore molto importante che regola la solubilità dell'ossigeno disciolto è l'intensità della radiazione solare. Da un lato questa variabile influenza la temperatura dell'acqua, e quindi anche in questo caso teoricamente la relazione dovrebbe essere inversamente proporzionale, poiché l'aumento della temperatura dell'acqua riduce la sua capacità di trattenere i gas disciolti. D'altra parte, le radiazioni solari sono il motore della fotosintesi effettuata dal fitoplancton, quindi un'intensità bassa porta a sottoproduzione di ossigeno e viceversa. Quest'ultimo effetto potrebbe prevalere sul primo, poiché è più diretto.
Si inseriscono anche i giorni con radiazioni solari particolarmente poco intense tra le holidays.
L'analisi del plot delle componenti dovrebbe rendere più chiara l'origine dei giorni forniti come holidays.

Livello dei fiumi

Si riporta innanzitutto lo storico dei dati del livello dei due fiumi rispetto a quello del mare tra marzo 2021 e novembre 2023 circa.

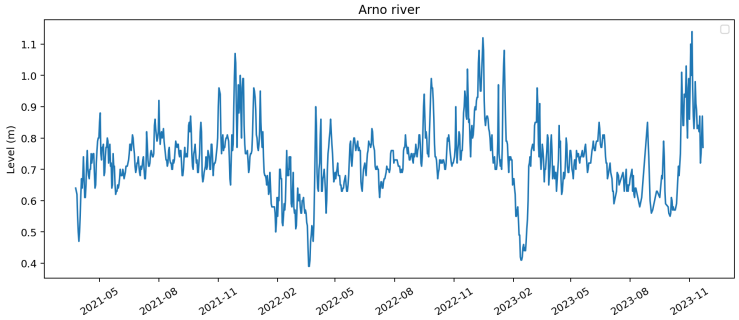


Figura 4.11: Livello del fiume Arno.

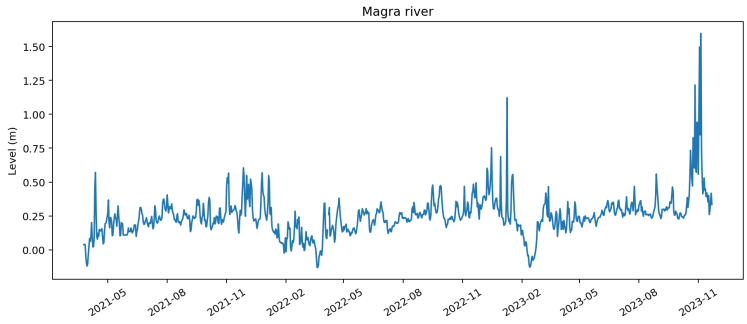


Figura 4.12: Livello del fiume Magra.

Il comportamento dei due fiumi risulta qualitativamente simile, con una concordanza tra minimi e picchi. Tuttavia, il Magra, essendo un fiume di dimensioni minori rispetto all'Arno, presenta oscillazioni di livello più contenute.
Per estrarre le date di rilevanza è stata creata una funzione apposita:

```
def get_holidays_from_river(df_river, start_train, end_forecast, title,
    percentage, plot_rivers = True, print_threshold = True,
    print_nr_days = True):
```

Listing 4.2: get_holidays_from_river function

Tale funzione riceve in input il dataframe contenente i livelli del fiume, lo taglia in base alle date desiderate e identifica un valore di livello che stabilisce se un giorno debba essre considerato holiday o meno. In particolare, il calcolo si basa sul percentile del livello del fiume: ad esempio, se il percentile indicato in input è del 90%, tutti i valori al di sopra di questa soglia verranno inseriti nel dataframe delle holidays, che verrà restituito come output. Tale dataframe in uscita rispetta il formato richiesto da Prophet: contiene una colonna con il nome dell'evento, una colonna con la data e due colonne che indicano gli estremi dell'intervallo di giorni prima e dopo l'evento da includere. In particolare, se il fiume considerato è l'Arno, si stima che l'effetto di una giornata di piena abbia influenza sulle rilevazioni in baia con un ritardo di 6 giorni, considerando delle correnti nel verso della baia, una velocità media dell'acqua di 0.1 m/s e una distanza in linea d'aria dalla baia di circa 50 km. Se invece le correnti non fossero nella direzione "giusta", il modello dovrebbe attribuire perso nullo a quella holiday.
Se invece si tratta del fiume Magra, data la vicinanza della foce alla baia, non si considera alcun lag temporale.
Infine, la funzione permette di visualizzare il plot del livello del fiume, evidenziando la relativa soglia selezionata, e di conoscere quanti sono i giorni categorizzati come holidays rispetto a quelli totali nel dataset.
Ad esempio, applicando la funzione al dataset del fiume Arno ristretto rispettivamente al mese di marzo e all'intero periodo disponibile, gli output ricevuti sono i seguenti:

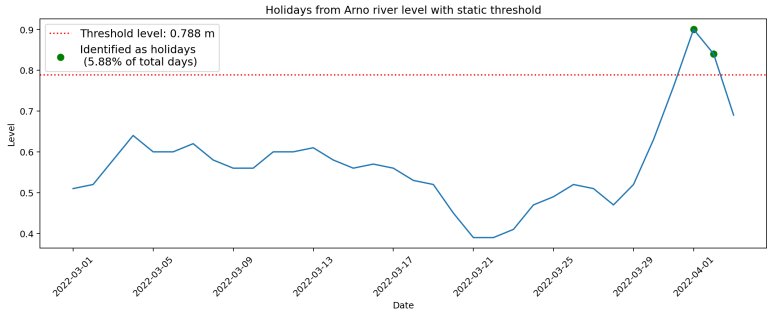


Figura 4.13: Holidays ottenute dal livello del fiume Arno per il mese di marzo 2022.

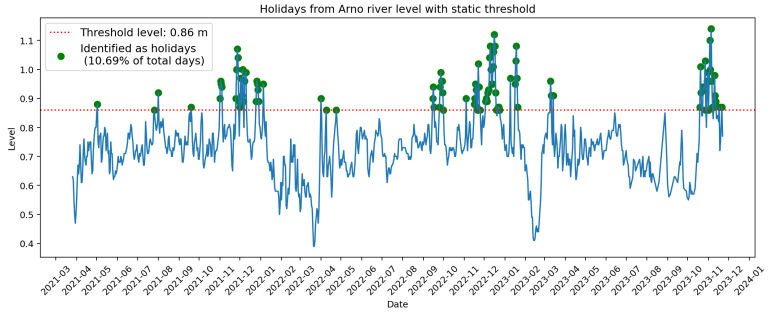


Figura 4.14: Holidays ottenute dal livello del fiume Arno con soglia statica per l'intero periodo disponibile.

Tuttavia, osservando il secondo grafico (4.14), si nota una potenziale perdita di informazioni. Questa funzione infatti non tiene conto dell'andamento stagionale della serie: le holidays vengono identificate quasi completamente ed esclusivamente durante il periodo invernale, che solitamente è caratterizzato da più pioggia e quindi livelli del fiume più alti, mentre i picchi che possono essersi verificati durante l'estate non sono tenuti in considerazione.
Per questo motivo, è stato necessario implementare una seconda funzione che prevedesse un livello threshold non statico, adatto alla stagionalità della serie.

```
def get_holidays_from_river_pro(df_river, start_train, end_forecast,
    title, threshold_factor = 1.05, plot_rivers = True, print_nr_days =
    True):
```

Listing 4.3: get_holidays_from_river_pro function

Questa funzione opera in generale come la precedente a livello di obiettivi, input e output, ma differisce per la definizione della soglia: la modellizzazione della serie stessa da parte di Prophet viene utilizzata nel corpo della funzione come livello dinamico. Una volta eseguito l'addestramento della serie (il forecast in questo caso è irrilevante, la finestra di previsione usata è nulla), viene estrapolata la colonna `yhat_upper`, ovvero il limite superiore dell'intervallo di incertezza stimato dal modello: questo livello, moltiplicato per

un piccolo fattore di aggiustamento, viene assunto come soglia adattativa rispetto all'evoluzione della serie. Tale limite è più significativo di una soglia fissa poiché comprende anche i termini di trend e di stagionalità annuale, la quale viene attivata nel caso in cui il periodo considerato sia maggiore di 6 mesi.
Anche in questo caso, considerando la lontananza del fiume Arno, il dataframe delle holidays restituito dalla funzione considera un lag temporale di 6 giorni per gli eventi relativi a tale fiume.
In seguito è riportato l'output che si ottiene, sempre per il fiume Arno (lo stesso ragionamento viene applicato per il fiume Magra) e per l'intero periodo disponibile:

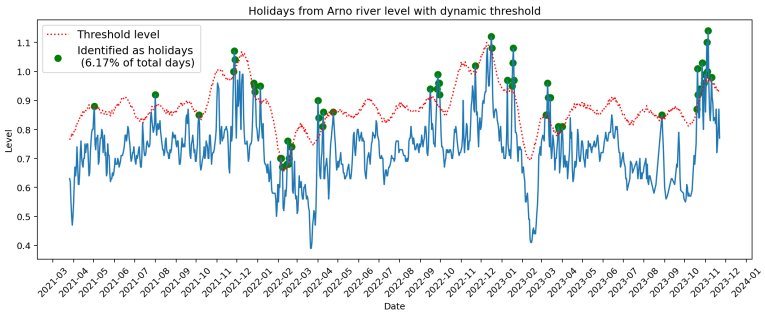


Figura 4.15: Holidays ottenute dal livello del fiume Arno con soglia dinamica per l'intero periodo disponibile.

Radiazioni solari

I dati delle radiazioni solari sono disponibili a cadenza oraria, quindi sono stati precedentemente raggruppati su base giornaliera. Si riporta la serie storica che è il risultato di tale operazione:

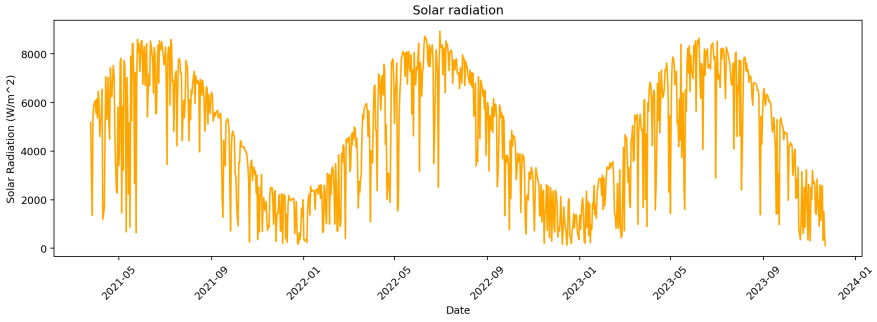


Figura 4.16: Radiazioni solari giornaliere per l'intero periodo disponibile.

Similarmente alla trattazione svolta nell'analisi del livello dei fiumi, anche per le radiazioni solari è stata creata una funzione che, a partire da un dataset e un certo range di date, estrae i giorni da considerare come holidays.

```
def get_holidays_from_radiation(df_radiation, start_train, end_forecast,
    threshold_factor = 0.9, plot_radiation = True, print_nr_days = True
   ):
```

Listing 4.4: get_holidays_from_radiation function

Per identificare la soglia dinamica di radiazione solare viene usata ancora una volta la previsione di Prophet, ma in questo caso, siccome i giorni di interesse sono quelli con radiazione solare particolarmente bassa, viene estratta la colonna `yhat_lower` invece che `yhat_upper`.
Si riporta l'output della funzione considerando l'intero periodo di dati a disposizione:

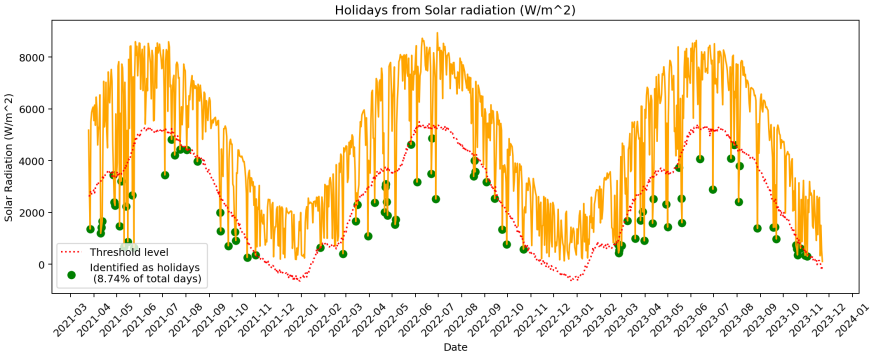
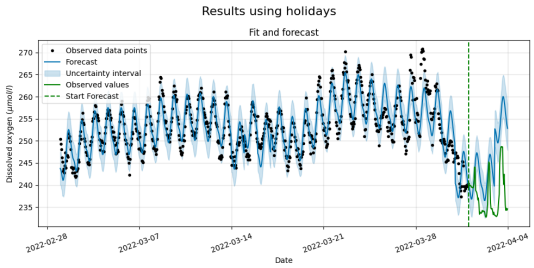


Figura 4.17: Holidays ottenute dalle radiazioni solari per l'intero periodo disponibile.

A questo punto, si possono concatenare i tre dataframe di holidays ottenuti (a partire dal livello del fiume Arno, del fiume Magra e dalle radiazioni solari) e fornirli alla funzione `fit_compare` per apprezzare le differenze nella previsione e nelle componenti della serie che vengono individuate.
Utilizzando il mese di marzo 2022 come training set e i primi tre giorni di aprile come test set, e adottando le impostazioni dei parametri che sono risultate migliori dalle analisi precdentemente svolte, si riportano il dataframe di holidays e i risultati in termini di plot ed errori:

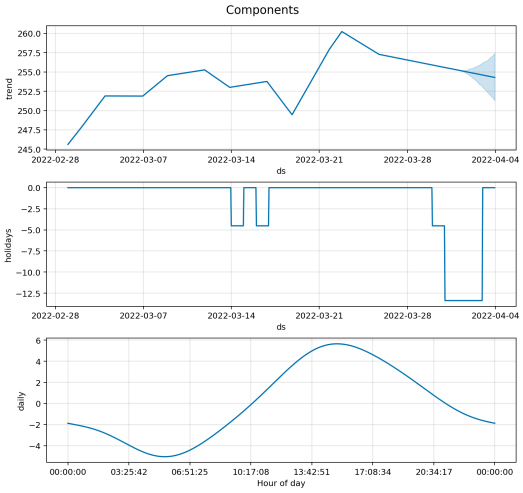
holiday	ds	lower_window	upper_window
portata alta Arno	2022-04-07	0	0
portata alta Arno	2022-04-08	0	0
portata alta Magra	2022-03-31	0	0
portata alta Magra	2022-04-01	0	0
portata alta Magra	2022-04-02	0	0
radiazione giornaliera bassa	2022-03-14	0	0
radiazione giornaliera bassa	2022-03-16	0	0
radiazione giornaliera bassa	2022-03-30	0	0

Tabella 4.7: Tabella relativa alle holidays da fornire a Prophet.



Metric	Train	Test	Total
MAE	2.202	8.641	2.770
MAPE (%)	0.868	3.652	1.114
RMSSE	1.525	5.510	2.191
mRMSSE	1.525	5.213	2.181

(a) Confronto tra valori reali ed errori ottenuti.

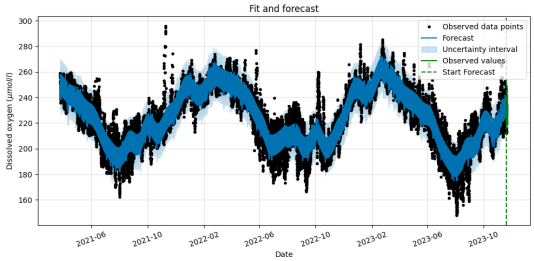


(b) Componenti della previsione.

Figura 4.18: Risultati ottenuti dal modello Prophet con holidays per marzo 2022.

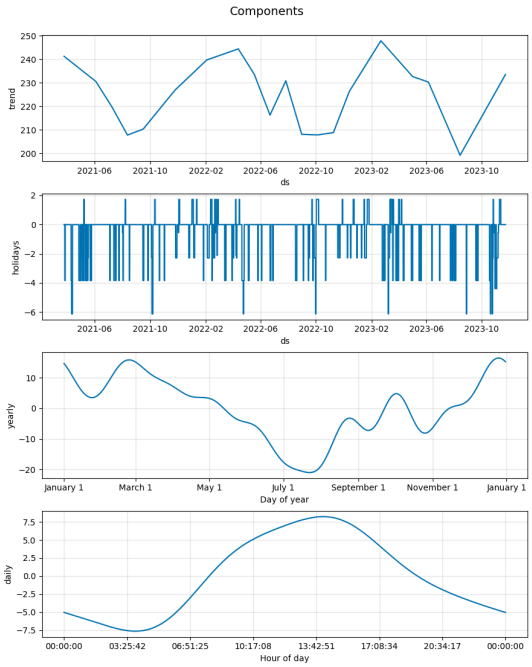
Si ricorda nuovamente che l'effetto dell'Arno è stimato essere in ritardo di 6 giorni rispetto a quando il livello è effettivamente più alto del solito. Essendo che i giorni di piena si verificano a fine marzo, come si è visto in figura 4.13, le holidays effettive cadono al di fuori anche del test set e quindi anche se sono presenti nel dataframe, non verranno considerate da Prophet.
In termini di precisione del modello, si può notare che gli errori di train diminuiscono lievemente, mentre quelli di test tendono ad aumentare. Come si vede dal plot delle componenti infatti e dal dataframe 4.7, le holidays che danno un contributo più significativo sono alla fine del periodo temporale, e questo va ad influire più pesantemente sulla pevisione e quindi sugli errori di test.
In particolare, l'apprendimento di Prophet ha portato a considerare le holidays come giorni che contribuiscono negativamente alla concentrazione di ossigeno disciolto, il che sembra contraddire le ipotesi iniziali discusse.
Tuttavia, l'arrivo in baia di molta acqua proveniente dai fiumi, non apporta solo acqua dolce, bensì anche materiale organico e residui dal terreno che vanno decomposti da batteri che consumano ossigeno e che intorbidiscono l'acqua, rendendo anche più difficile la fotosintesi da parte del fitoplancton.
Invece per quanto riguarda i giorni con radiazioni solari particolarmente basse, individuate intorno a metà mese, è vero che la temperatura diminuisce e quindi l'ossigeno dovrebbe essere più solubile, tuttavia anche in questo caso la fotosintesi viene inibita e quindi la concentrazione di ossigeno è ridotta.

Si procede ora con la stessa analisi dei risultati tenendo conto delle holidays durante tutto il periodo disponibile, seguita da un plot zoomato sulla previsione:



Metric	Train	Test	Total
MAE	6.69	8.30	6.70
MAPE (%)	3.06	3.71	3.06
RMSSE	2.55	2.98	2.55
mRMSSE	2.55	6.903	2.55

(a) Confronto tra valori reali e previsti ed errori ottenuti.



(b) Componenti della previsione.

Figura 4.19: Risultati ottenuti dal modello Prophet con holidays per l'intero periodo disponibile.

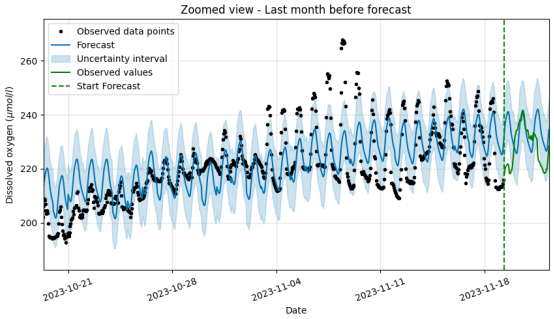


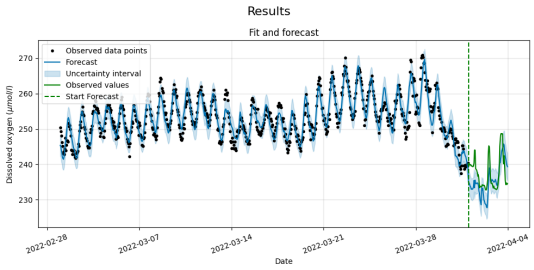
Figura 4.20: Zoom del confronto tra reali e previsti ottenuti dal modello Prophet con holidays per l'intero periodo disponibile.

Si ottiene lo stesso comportamento anche in questo caso: le holidays non danno un contributo davvero significativo, gli errori di train riusltano essere leggermente più bassi mentre quelli di test più alti, risultando in un leggero miglioramento sulla media dei due periodi.
In questa seconda analisi si osserva che, nello specifico contesto marino in cui vengono prese le misurazioni, i giorni con livelli fluviali eccezionalmente alti e quelli con bassa radiazione solare sono stati associati talvolta a un contributo negativo e talvolta positivo sulla concentrazione di ossigeno disciolto. Questo fenomeno rivela la complessità dell'ecosistema marino in esame,

suggerendo ancora una volta l'importanza di tutti i processi biochimici e fisici coinvolti.
L'approccio basato sulle holidays si è dimostrato più efficace nella regressione della serie, ma ha un limite intrinseco: esso considera questi eventi come fenomeni binari (presenti oppure assenti) senza tenere conto della loro portata. Un picco nei livelli fluviali ha lo stesso peso di un picco più moderato, così come per i minimi delle radiazioni solari.
Per superare questa limitazione e sfruttare appieno le informazioni dei dati, il presente studio passerà a un'integrazione più completa. La prossima sezione esplorerà come si comporta il modello di Prophet quando le serie storiche dei livelli fluviali e delle radiazioni solari vengono aggiunte non più come holidays binarie, ma come regressori esterni.

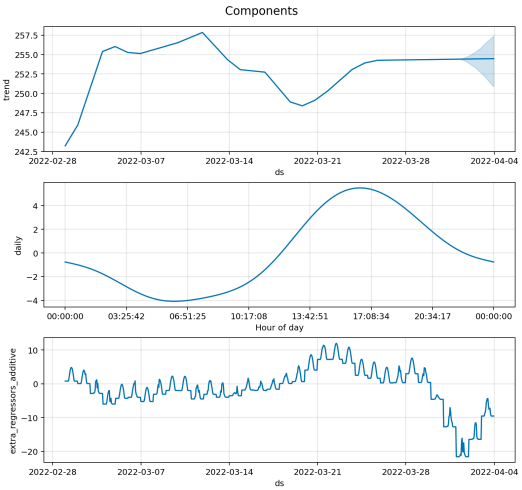
4.5.3 Introduzione dei regressori

Questo approccio permette a Prophet di imparare una relazione continua e proporzionale tra le variabili fornite come regressori e l'ossigeno disciolto. Invece di limitarsi a identificare un evento che si discosta dalla norma, il modello può correlare l'intensità della portata dell'evento con l'entità della variazione dell'ossigeno disciolto, fornendo una previsione più precisa.
I regressori vanno forniti alla funzione che implementa Prophet tramite un dataframe.
Come prima, si riportano i risultati relativi al periodo di marzo 2022:



Metric	Train	Test	Total
MAE	1.834	4.043	2.029
MAPE (%)	0.724	1.692	0.809
RMSSE	1.214	2.418	1.364
mRMSSE	1.214	2.288	1.358

(a) Metriche di valutazione su train, test e totale.

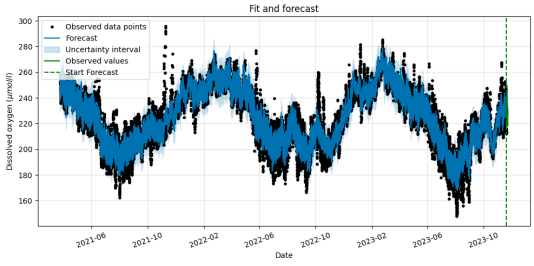


(b) Componenti della previsione.

Figura 4.21: Risultati ottenuti dal modello Prophet multivariato per marzo 2022.

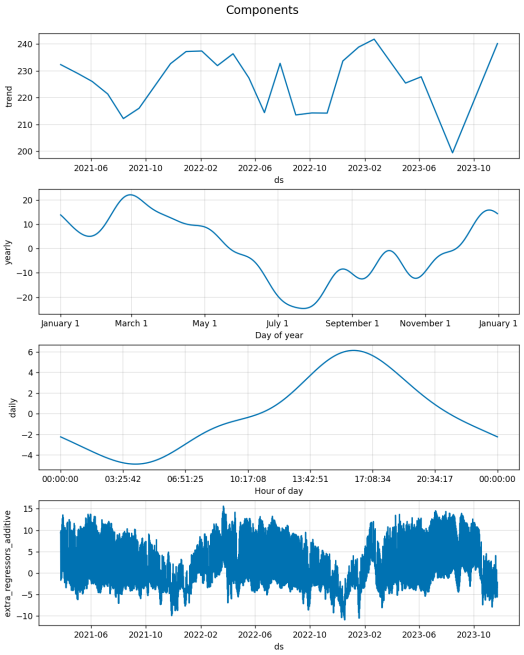
Rispetto ai risultati ottenuti senza considerare informazioni aggiuntive (nè holidays nè regressori), ovvero quelli riportati nella tabella 4.18a, c'è un miglioramento di ogni metrica per ogni periodo. Le differenze non sono così tanto apprezzabili ed evidenti poiché il modello di base di partenza era già abbastanza robusto.
Ciò che salta all'occhio è la differenza nel plot delle componenti. Non si ha più un grafico 'a salti' in date specifiche come nel caso dell'inclusione delle holidays, adesso il grafico relativo ai regressori additivi segue a grandi linee la curva della serie di partenza, con piccole oscillazioni giornaliere. Questo indica che il modello ha imparato una relazione graduale e continua tra

ossigeno e regressori ed è abbastanza sensibile da catturare le fluttuazioni quotidiane della radiazione solare e di incorporarle nella previsione.
Di seguito i risultati relativi all'intero periodo disponibile:



Metric	Train	Test	Total
MAE	5.264	5.285	5.264
MAPE (%)	2.389	2.373	2.389
RMSSE	2.079	1.958	2.079
mRMSSE	2.079	4.525	2.082

(a) Confronto tra valori reali e previsti ed errori ottenuti.



(b) Componenti della previsione.

Figura 4.22: Risultati ottenuti dal modello Prophet multivariato per l'intero periodo disponibile.

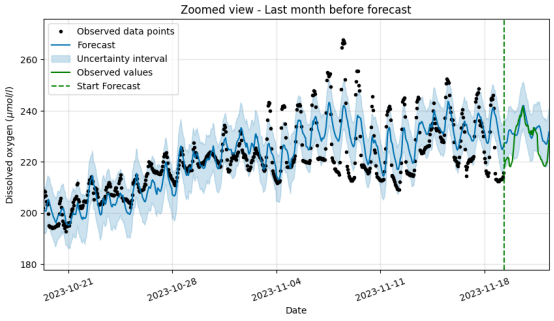


Figura 4.23: Zoom del confronto tra reali e previsti ottenuti dal modello Prophet multivariato per l'intero periodo disponibile.

Ancora una volta, i risultati sono migliori rispetto al caso base e al caso che include le holidays.

Capitolo 5

Risultati

In questo capitolo saranno esplorati i risultati dei modelli SARIMA e Prophet, nei casi univariato, multivariato e con aggiunta di holidays nel caso di Prophet, con l'obiettivo di confrontarli e approfondire vantaggi e svantaggi dei due metodi.
Per ottenere una valutazione più robusta delle prestazioni dei modelli implementati, invece che basarsi su un'analisi limitata su singoli periodi di previsione, come ad esempio gli ultimi 3 giorni disponibili del dataset, usati in precedenza, è stata implementata una rolling forecast evaluation. In particolare, l'intero periodo temporale disponibile (di circa due anni e mezzo) è stato suddiviso in sotto-intervalli temporali: per ciascuno di essi, il modello è stato addestrato utilizzando due anni consecutivi come train set mentre è stato testato sui tre giorni immediatamente successivi. Ad ogni iterazione, la finestra di train (e di conseguenza quella di test) viene traslata in avanti di tre giorni, in modo che i giorni di test non si sovrappongano mai, garantendo così che ogni previsione venga valutata su dati completamente nuovi e indipendenti. Il processo viene ripetuto fino ad arrivare all'ultima data disponibile, salvando gli errori di test e i tempi necessari per l'addestramento del modello.
La scelta di mantenere due anni di dati di addestramento per ciascuna finestra è motivata dalla necessità di fornire al modello almeno due cicli stagionali completi da utilizzare in fase di training, in modo da catturare correttamente le dinamiche annuali della concentrazione dell'ossigeno. Questa necessità è specifica del modello Prophet, che anche nelle sue impostazioni di default, imposterebbe la stagionalità annuale solo nel caso di disponibilità di due cicli annuali completi di dati [10]. Tuttavia, oltre che essere consigliabile per avere un confronto pari, anche per il modello SARIMA$(2,1,2)(1,1,2)_{24}$ è utile avere molti dati a disposizione. Infatti, nonostante la previsione immediata dipenda al massimo dal dato di 48 ore prima (a causa dell'ordine stagionale $Q = 2$), un ampio set di training è comunque necessario per stimare al meglio i parametri del modello (i coefficienti autoregressivi, di media mobile stagionali e non, e la varianza del rumore bianco).
Questo approccio con rolling window consente di valutare i modelli in condizioni differenti di stagionalità, seppure parzialmente: infatti, esclusi i primi due anni di dati (dal 25 marzo 2021 al 25 marzo 2023), rimane il periodo compreso tra il 25 marzo 2023 e il 22 novembre 2023, e quindi non viene mai performata una previsione durante il periodo invernale. Tuttavia, tale valutazione garantisce che le metriche non dipendano da un particolare periodo dell'anno o da eventi anomali in fase di addestramento.
Per la valutazione, viene calcolata la media sulle metriche di test ottenuti in ciascuna finestra, così da fornire una stima complessiva dell'accuratezza del modello sull'intero periodo di analisi. Inoltre, ad ogni iterazione viene salvato il tempo computazionale necessario per addestramento e previsione, calcolato tramite `tic,toc` della libreria `time`.

5.1 SARIMA

5.1.1 Modello univariato

Si riportano la testa e la coda della tabella di errori ottenuti in fase di previsione relativi ai singoli step, corrispondenti alla data di inizio forecast. Successivamente, nella tabella 5.2 sono presentati i risultati mediati sull'intero periodo.

Date	MAE	MAPE (%)	RMSSE	mRMSSE	CRPS	Time (s)
2023-03-25	4.856	1.983	1.704	3.229	4.932	272.407
2023-03-28	4.017	1.626	1.440	2.043	4.822	279.060
2023-03-31	4.987	2.090	1.776	2.847	5.142	267.496
2023-04-03	7.956	3.294	2.652	4.581	5.853	270.995
2023-04-06	4.199	1.695	1.882	2.005	5.145	286.282
⋮	⋮	⋮	⋮	⋮	⋮	⋮
2023-11-05	5.508	2.334	2.059	1.532	4.202	291.699
2023-11-08	11.587	5.136	3.632	2.162	7.729	377.187
2023-11-11	6.943	3.013	2.602	1.858	4.824	377.699
2023-11-14	6.212	2.709	2.079	2.244	4.078	344.198
2023-11-17	8.569	3.808	2.870	2.704	5.534	323.025

Tabella 5.1: Metriche calcolate ad ogni step della valutazione rolling per il modello SARIMA univariato.

MAE	7.87
MAPE (%)	3.72
RMSSE	2.67
mRMSSSE	2.80
CRPS	6.10
Time (s)	312.09

Tabella 5.2: Valori medi delle metriche della valutazione rolling per il modello SARIMA univariato.

Gli errori ottenuti con il modello SARIMA univariato risultano nel complesso soddisfacenti: un valore di MAE di circa 8 $\mu mol/l$ a fronte di un valore medio dei dati osservati di 250 $\mu mol/l$ è considerabile contenuto. Anche il MAPE inferiore al 5 % indica una buona capacità del modello di prevedere l'andamento della serie temporale.
Tuttavia, il tempo computazionale medio di 312 secondi, quindi circa 5 minuti per step, è pittosto elevato, segno di una complessità non trascurabile del processo di stima.
Poiché mediando i risultati si perde parte dell'informazione, si riporta di seguito l'istogramma della distribuzione del MAE, per poter apprezzare anche la magnitudo e la frequenza di tali valori della metrica.

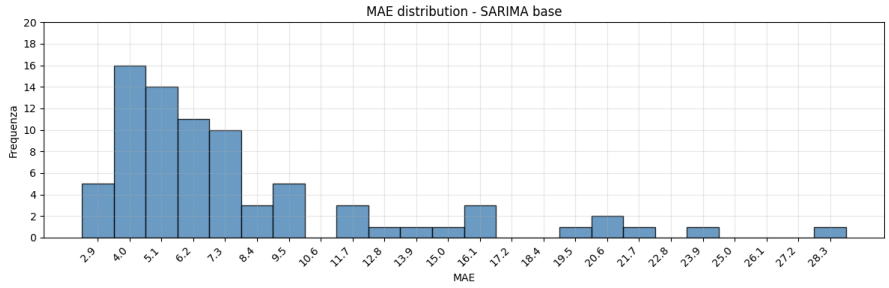


Figura 5.1: Distribuzione del MAE nella valutazione rolling del modello SARIMA univariato.

Dalla distribuzione si evince che la maggior parte dei valori si concentra tra 4 e 5 $\mu mol/l$, con una coda verso destra che riflette alcuni casi di errori più elevati, associabili a momenti di brusche variazioni della serie, in cui il moello fatica a prevedere la dinamica seguita dall'ossigeno.

5.1.2 Modello multivariato

Come nel caso univariato, vengono riportati i risultati della valutazione rolling del modello SARIMA esteso con regressori esogeni.

Date	MAE	MAPE (%)	RMSSE	mRMSSE	CRPS	Time (s)
2023-03-25	4.745	1.935	1.666	3.158	4.610	1104.530
2023-03-28	4.356	1.750	1.540	2.185	4.750	1181.180
2023-03-31	3.963	1.661	1.462	2.345	4.690	1161.510
2023-04-03	5.013	2.065	1.718	2.967	5.110	1151.830
2023-04-06	5.732	2.334	2.161	2.302	5.230	1137.770
⋮	⋮	⋮	⋮	⋮	⋮	⋮
2023-11-05	8.377	3.689	2.672	1.988	5.930	1209.700
2023-11-08	12.413	5.541	3.889	2.315	8.640	1120.260
2023-11-11	4.339	1.910	1.565	1.118	3.140	1106.780
2023-11-14	4.971	2.131	1.789	1.930	3.590	1089.400
2023-11-17	7.855	3.541	2.773	2.612	5.370	1109.960

Tabella 5.3: Metriche calcolate ad ogni step della valutazione rolling per il modello SARIMA multivariato.

MAE	7.64
MAPE (%)	3.64
RMSSE	2.59
mRMSSSE	2.66
CRPS	6.36
Time (s)	1174.79

Tabella 5.4: Valori medi delle metriche della valutazione rolling per il modello SARIMA multivariato.

Tutte le metriche di errore, ad eccezione del CRPS, risultano leggermente diminuite rispetto al caso univariato dello stesso modello. Tuttavia, la riduzione di MAE e mRMSSE (rispettivamente da 7.87 a 7.64 e da 2.67 a 2.59) indica un miglioramento davvero marginale, che non giustifica l'enorme aumento del costo computazionale: il tempo medio per step è salito a 1175 secondi, ovvero circa 20 minuti, quadruplicando rispetto al modello base.
Di seguito si riporta la distribuzione del MAE, per vedere se l'intervento dei regressori abbia modificato la variabilità degli errori.

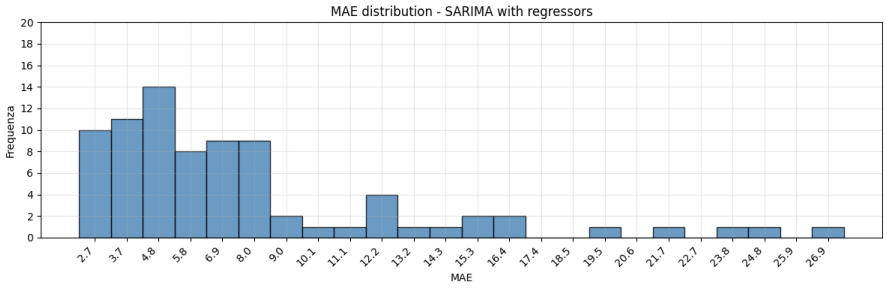


Figura 5.2: Distribuzione del MAE nella valutazione rolling del modello SARIMA multivariato.

Effettivamente, l'istogramma mostra uno spostamento della distribuzione verso sinistra, dove la maggior parte dei valori si attesta tra 2.7 e 4.8, segno che in diversi casi il modello riesce a produrre previsioni più precise. Tuttavia, la presenza di una porzione ancora consistente di errori superiori a 6 $\mu mol/l$ conferma che la stabilità del modello non è sempre garantita.

Errori di previsione e andamento dell'ossigeno nel tempo

Si procede con un confronto tra i valori di MAE ottenuti (in base alla data di inizio della previsione) applicando il modello SARIMA univariato e multivariato, in relazione all'andamento della variabile di interesse, ovvero la concentrazione di ossigeno disciolto. L'obiettivo è quello di identificare eventuali correlazioni tra l'errore commesso dai modelli e i momenti di maggiore instabilità nei dati.

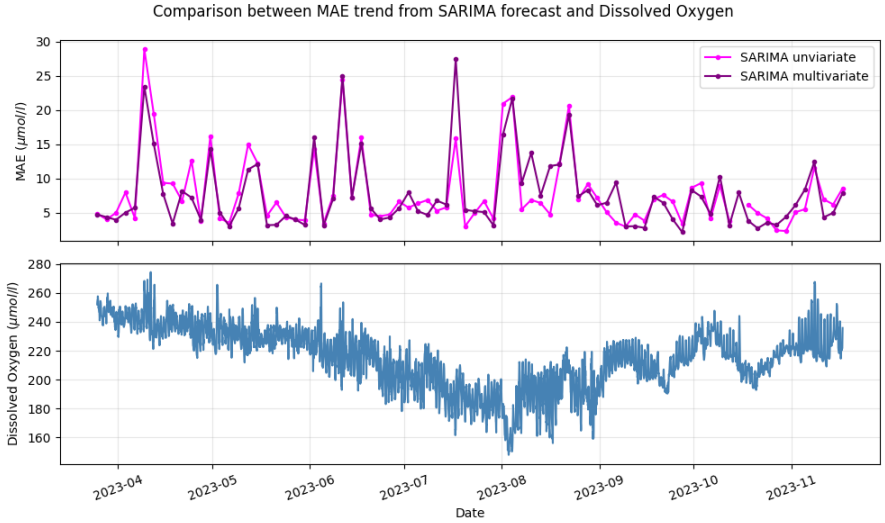


Figura 5.3: Confronto tra andamento del MAE (per data di inizio previsione) nella valutazione rolling del modello SARIMA e andamento dell'ossigeno.

Il grafico mostra una forte variabilità del MAE nel tempo, con alcuni picchi evidenti che, nella maggior parte dei casi, corrispondono a periodi di maggiore instabilità nella serie dell'ossigeno disciolto. In particolare, gli incrementi del MAE si osservano in corrispondenza di fasi in cui l'ossigeno disciolto presenta variazioni rapide, che siano dei bruschi incrementi oppure diminuizioni rispetto ai valori precedenti.
Nel complesso, il modello SARIMA multivariato presenta valori di errore generalmente inferiori rispetto al caso univariato: le prestazioni peggiorano per le previsioni che iniziano nel periodo estivo, quando le condizioni ambientali sono più variabili.

5.2 Prophet

5.2.1 Modello univariato

Si riportano la testa e la coda della tabella degli errori ottenuti in ciascuno step, corrispondente alla data di inizio forecast, e i tempi necessari per l'addestramento. Successivamente, nella tabella 5.2 sono presentate le metriche medie calcolate sull'intero periodo di analisi.

Date	MAE	MAPE (%)	RMSSE	mRMSSE	Time (s)
2023-03-25	3.971	1.616	1.461	2.769	10.663
2023-03-28	4.151	1.675	1.475	2.093	11.734
2023-03-31	4.607	1.918	1.617	2.592	13.541
2023-04-03	3.306	1.347	1.289	2.227	14.082
2023-04-06	3.009	1.219	1.550	1.651	10.251
⋮	⋮	⋮	⋮	⋮	⋮
2023-11-05	7.652	3.264	2.690	2.001	14.573
2023-11-08	16.084	7.095	5.061	3.013	17.756
2023-11-11	11.656	5.353	4.201	3.000	13.497
2023-11-14	6.127	2.689	2.377	2.564	13.155
2023-11-17	10.575	4.785	3.736	3.519	16.034

Tabella 5.5: Metriche calcolate ad ogni step della valutazione rolling per il modello Prophet univariato.

MAE	8.77
MAPE (%)	4.21
RMSSE	2.95
mRMSSE	3.41
Time (s)	13.06

Tabella 5.6: Valori medi delle metriche della valutazione rolling per il modello Prophet univariato.

Rispetto ai modelli SARIMA, i risultati di Prophet univariato mostrano valori più elevati: ad esempio il MAE aumenta di un'unità e il MAPE sale al 4.21%, contro il 3.72% del modello precedente.
Tuttavia, il tempo computazionale medio impiegato da Prophet è significativamente più basso (13 secondi contro 312 secondi del modello SARIMA univariato), rendendolo più efficiente.
Inoltre, la distribuzione del MAE è rappresentata nel grafico seguente, che mostra la magnitudo e la frequenza degli errori.

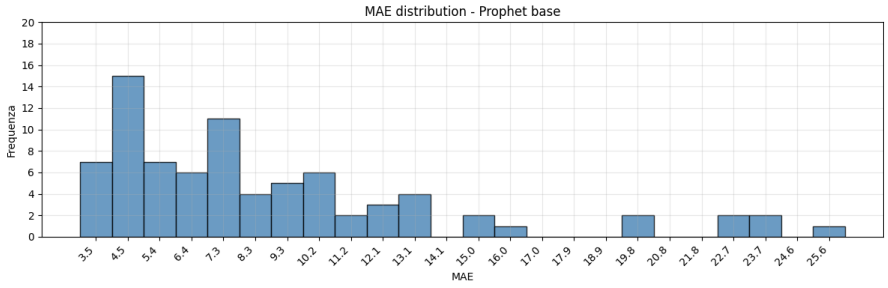


Figura 5.4: Distribuzione del MAE nella valutazione rolling del modello Prophet univariato.

La distribuzione presenta un picco significativo relativo al valore di MAE di 4.5 $\mu mol/l$, seguito da un altro picco minore di valore pari a 7.3 circa.

5.2.2 Modello con holidays

Nel caso del modello Prophet con aggiunta di holidays, trattate nella sezione 4.5.2, i risultati mostrano un leggero miglioramento rispetto alla versione univariata, come evidenziato nelle tabelle sottostanti. La tabella 5.7 mostra le metriche calcolate per ciascuno step, mentre la tabella 5.9 riportano la sintesi dei risultati medi.

Date	MAE	MAPE (%)	RMSSE	mRMSSE	Time (s)
2023-03-25	3.617	1.468	1.278	2.423	20.115
2023-03-28	4.655	1.870	1.748	2.479	14.909
2023-03-31	4.760	1.983	1.672	2.682	15.194
2023-04-03	3.513	1.435	1.364	2.355	14.197
2023-04-06	3.508	1.424	1.799	1.916	17.838
⋮	⋮	⋮	⋮	⋮	⋮
2023-11-05	7.538	3.211	2.646	1.969	16.749
2023-11-08	16.554	7.333	5.194	3.092	19.327
2023-11-11	11.904	5.465	4.324	3.088	15.509
2023-11-14	5.782	2.534	2.304	2.486	14.601
2023-11-17	11.201	5.071	3.931	3.703	13.536

Tabella 5.7: Metriche calcolate ad ogni step della valutazione rolling per il modello Prophet con holidays.

MAE	8.67
MAPE (%)	4.14
RMSSE	2.94
mRMSSE	3.38
Time (s)	16.51

Tabella 5.8: Valori medi delle metriche della valutazione rolling del modello Prophet con holidays.

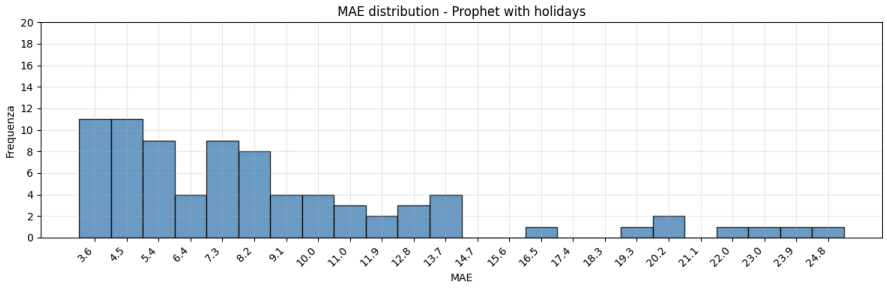


Figura 5.5: Distribuzione del MAE nella valutazione rolling del modello Prophet con holidays.

Tale variante mostra risultati complessivamente molto simili a quelli del modello univariato, con un miglioramento non significativo. Il tempo computazionale per step di previsione, invece, è leggermente aumentato, a causa dell'inclusione delle festività.

5.2.3 Modello multivariato

Infine, si riportano i risultati relativi al modello Prophet con aggiunta di regressori.

Date	**MAE**	**MAPE (%)**	**RMSSE**	**mRMSSE**	**Time (s)**
2023-03-25	3.443	1.389	1.256	2.381	17.963
2023-03-28	5.421	2.207	1.936	2.746	22.967
2023-03-31	7.153	2.976	2.403	3.853	20.743
2023-04-03	4.747	1.932	1.699	2.935	17.987
2023-04-06	3.469	1.423	1.466	1.562	18.910
⋮	⋮	⋮	⋮	⋮	⋮
2023-11-05	10.298	4.562	3.313	2.465	13.008
2023-11-08	12.452	5.410	4.135	2.462	15.625
2023-11-11	9.389	4.179	2.958	2.113	18.594
2023-11-14	5.211	2.264	2.082	2.247	19.363
2023-11-17	8.909	3.987	3.071	2.892	15.947

Tabella 5.9: Metriche calcolate ad ogni step della valutazione rolling per il modello Prophet multivariato.

MAE	6.68
MAPE (%)	3.17
RMSSE	2.31
mRMSSE	2.54
Time (s)	18.19

Tabella 5.10: Valori medi delle metriche della valutazione rolling per il modello Prophet multivariato.

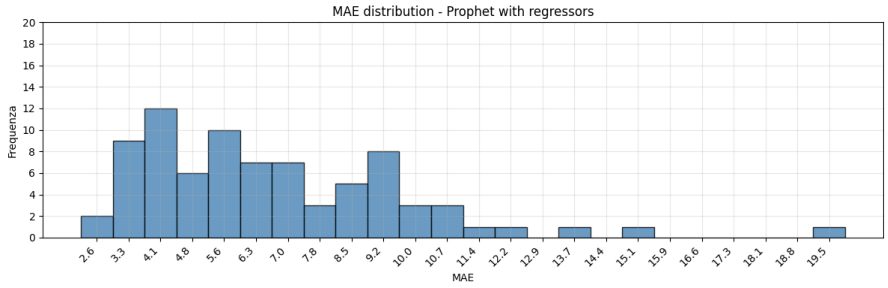


Figura 5.6: Distribuzione del MAE nella valutazione rolling del modello Prophet multivariato.

L'introduzione di regressori esogeni ha migliorato ulteriormente le performance del modello, passando da un MAE di 8.7 circa dei modelli precedenti a 6.7, classificandosi come il modello migliore nella previsione della concentrazione di ossigeno disciolto.
Il tempo computazionale è lievemente aumentato a causa della maggiore complessità del modello (18 secondi contro 13 secondi per il modello di base), ma rimane comunque molto inferiore rispetto al tempo richiesto dal modello SARIMA.

Errori di previsione e andamento dell'ossigeno nel tempo

Come per la trattazione del moello SARIMA, nel grafico seguente viene mostrato il confronto tra l'andamento del MAE (per data di inizio previsione) e l'andamento dell'ossigeno disciolto, per ogni versione del modello Prophet.

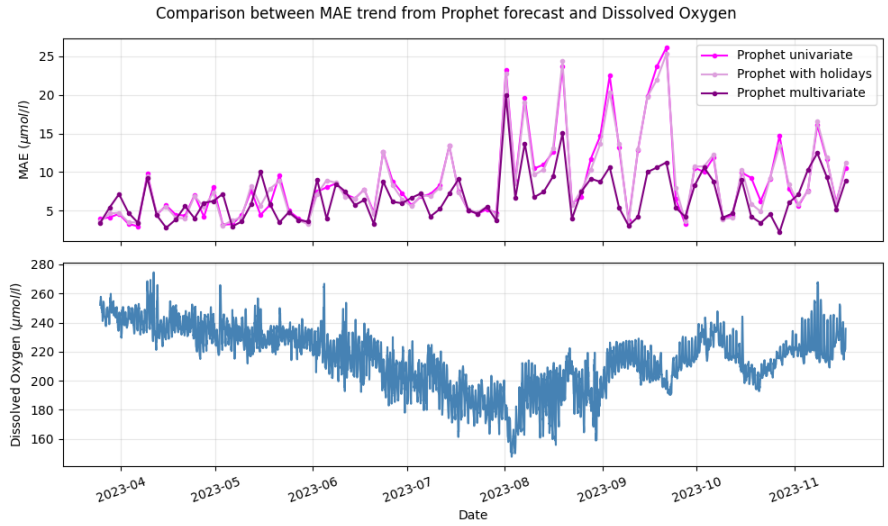


Figura 5.7: Confronto tra andamento del MAE (per data di inizio previsione) nella valutazione rolling del modello Prophet e andamento dell'ossigeno.

Il grafico mostra una maggiore stabilità dell'andamento del MAE rispetto ai risultati del modello SARIMA in figura 5.3, ad eccezione del periodo estivo. I picchi relativi alle previsioni che

iniziano ad agosto e settembre corrispondono a periodi di maggiore variabilità dell'ossigeno, dovuta probabilmente a fattori biologici e ambientali.
È altresì possibile che parte di questa dinamica sia influenzata dal valore fissato del parametro `changepoint_range = 0.9`, che limita la possibilità del modello di aggiornare rapidamente il trend in presenza di variazioni improvvise sulla fine del periodo di addestramento. Tuttavia, questa impostazione è stata considerata necessaria per non generare overfitting.
Si osserva inoltre che il modello con le festività offre delle prestazioni del tutto simili al modello univariato, mentre la versione multivariata di Prophet mostra un andamento del MAE più stabile e dei valori medi inferiori, specialmente nel periodo estivo, in cui c'è maggiore discrepanza: questo indica una migliore capacità predittiva e l'efficacia dell'introduzione dei regressori, come confermato dalle tabelle delle metriche medie.

Capitolo 6

Conclusioni e sviluppi futuri

Questa trattazione ha avuto come obiettivo quello di testare le prestazioni di modelli statistici volti a prevedere l'andamento della concentrazione di ossigeno nella baia di Santa Teresa di Lerici, al fine di monitorare lo stato di salute della baia.
La ricerca condotta in questa tesi è partita da un'analisi dei dati a disposizione, che si presentano come serie temporali: si tratta infatti di misurazioni a cadenza oraria di parametri marini e metereologici: concentrazione di ossigeno disciolto, temperatura dell'acqua, salinità, intensità di radiazioni solari, precipitazioni e così via. Osservando l'andamento dell'ossigeno disciolto, la variabile target, è emersa una forte stagionalità di tipo sia giornaliero che annuale. La prima è dovuta al ciclo solare giorno-notte, che stimola o inibisce la fotosintesi da parte del fitoplancton presente nella baia. La seconda è dovuta alla relazione con la temperatura dell'acqua, che varia in base alla stagione: nei mesi estivi è maggiore, e questo causa una diminuzione della solubilità dell'ossigeno, nei mesi più freddi invece la concentrazione del gas aumenta.
L'importanza della componente stagionale ha suggerito l'utilizzo di modelli capaci di trattare tale caratteristica. I metodi che sono stati utilizzati per la modellizzazione e la previsione dell'ossigeno disciolto sono stati due: il modello SARIMA, di ordini $(p, d, q)(P, D, Q) = (2,1,2)(1,1,2)$ e parametro di stagionalità pari a 24 (ore), e il modello Prophet, i cui parametri sono stati impostati in modo da modellizzare una serie con stagionalità giornaliera e annuale.
Entrambi i metodi sono stati testati nella loro versione univariata e multivariata, ovvero con l'aggiunta di regressori esterni, rappresentati dagli altri parametri, marini e metereologici. Prophet è stato anche testato nella sua versione con holidays: infomazioni binarie slla presenza di eventi anomali, come i giorni di piena dei fiumi Arno e Magra che sfociano nelle vicinanze della baia, e i giorni con radiazioni solari totali basse.
L'accuratezza delle previsioni è stata testata tramite il calcolo di alcune metriche come il Mean Absolute Error (MAE), il Mean Absolute Percentage Error (MAPE), una versione modificata del Root Mean Squared Scaled Error (RMSSE) e il Continuous Ranked probability Score (CRPS) nel caso del modello SARIMA, che restituisce previsioni probabilistiche.
La valutazione è stata implementata tramite una rolling forecast evaluation, ovvero utilizzando 2 anni di dati per l'addestramento e i 3 giorni successivi per la previsione, traslando di volta in volta fino a coprire tutto il dataset.
I risultati mediati su tutte le finestre di valutazione sono stati molto simili tra i due modelli e nel complesso soddisfacenti. Gli errori medi assoluti (in $\mu mol/l$) nel caso univariato sono di 7.9 per SARIMA e di 8.7 per Prophet, che risultano in un MAPE del 3.7% e del 4.2%, rispettivamente. L'aggiunta di holidays al modello Prophet porta a un miglioramento non significativo di 0.1 $\mu mol/l$.

Nel caso multivariato, fornendo regressori ai modelli, si passa a un MAE di 7.6 per SARIMA: vi è quindi un miglioramento minimo, anche se la distribuzione degli errori si sposta verso sinistra, ovvero la frequenza di errori più contenuti diventa più alta rispetto al caso univariato. L'errore commesso da Prophet, invece, scende a 6.7, risultando l'approccio più performante tra quelli testati.
La motivazione per cui Prophet riesce a migliorare con l'ausilio dei regressori, mentre SARIMA in modo meno significativo, potrebbe risiedere nella loro struttura. Il modelo Prophet aggiunge il contributo dei regressori in una componente lineare distinta dalla parte del trend e della stagionalità, che quindi può trattare in modo separato e flessibile. Nel caso del modello SARIMAX, invece, i regressori vengono integrati direttamente nella struttura autoregressiva e di media mobile del processo: i coefficienti associati ai regressori vengono stimati insieme ai parametri delle componenti AR e MA, all'interno di un'unica massimizzazione della verosimiglianza. Tale approccio comporta un aumento della complessità del modello: il numero di parametri da stimare cresce e le dipendenze tra essi possono rendere la stima meno stabile, soprattutto in presenza di rumore o correlazione tra i regressori.
Un aspetto molto rilevante che è emerso durante l'analisi, oltre alle metriche di errore, è la differenza di costi computazionali tra i due metodi. Se Prophet permette di ottenere previsioni in meno di mezzo minuto, SARIMA impiega circa 5 minuti nel suo caso base, e quasi 20 minuti nel caso in cui debba integrare le variabili esogene, risultando poco conveniente se si ha necessità di avere previsioni dei dati della baia quasi in tempo reale, come nel caso del progetto SmartTwin.

Sebbene i modelli statistici classici come SARIMA e Prophet abbiano fornito risultati soddisfacenti nel contesto analizzato, un possibile sviluppo futuro della trattazione riguarda l'impiego di metodi basati sul machine learning e di approcci ibridi, che combinino la capacità descrittiva dei modelli tradizionali con la flessibilità degli algoritmi di apprendimento automatico, o che combinino approcci di machine learning tra loro. Ad esempio, si potrebbe pensare alla costruzione di modelli ibridi SARIMA–ML, in cui il modello statistico cattura la struttura lineare della serie temporale, mentre l'algoritmo di machine learning apprende le componenti residue non lineari.
Tra gli algoritmi più efficaci in questo ambito spicca LightGBM (Light Gradient Boosting Machine), un modello basato su alberi decisionali e quindi caratterizzato da tempi di addestramento molto ridotti. Questo algoritmo, sviluppato da Microsoft Research, si è distinto come vincitore della competition M5 Forecasting di Kaggle, grazie alla sua capacità di catturare pattern complessi nei dati temporali. La sua struttura consente infatti di modellizzare interazioni non lineari tra regressori e variabile target, risultando particolarmente adatta per applicazioni multivariate o con numerosi regressori esogeni.
Un'ulteriore estensione potrebbe riguardare l'impiego di modelli di deep learning, come le reti neurali ricorrenti (RNN) o le Long Short-Term Memory (LSTM), che sono in grado di apprendere dipendenze temporali di lungo periodo. Tuttavia, questi approcci richiedono dataset di dimensioni maggiori e un tuning più complesso dei parametri.
In definitiva, questo lavoro si inserisce nel più ampio obiettivo di applicare metodi avanzati alla comprensione della complessità dei sistemi ambientali, rappresentando una base su cui costruire ulteriori sviluppi di ricerca e applicazioni operative.

Bibliografia

[1] Aayush Bajaj. Arima and sarima: Real-world time series forecasting guide. https://neptune.ai/blog/arima-sarima-real-world-time-series-forecasting-guide, 2023.

[2] Vito Ferri, Sele Okeoghene Thomas, Andrea Bordone, Giancarlo Raiteri, Tiziana Ciuffardi, Chiara Lombardi, Chiara Petrioli, Daniele Spaccini, Petrika Gjanci, Francesca Pennecchi, Marco Coisson, and Gianfranco Durin. A multi-stage model for dissolved oxygen monitoring of coastal seawater. In *2024 IEEE International Workshop on Metrology for the Sea; Learning to Measure Sea Health Parameters (MetroSea)*, pages 501–506. IEEE, October 2024. doi: 10.1109/metrosea62823.2024.10765778.

[3] Fondriest Environmental, Inc. Dissolved oxygen, November 2013. URL https://www.fondriest.com/environmental-measurements/parameters/water-quality/dissolved-oxygen/.

[4] Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437. URL https://sites.stat.washington.edu/raftery/Research/PDF/Gneiting2007jasa.pdf.

[5] James D. Hamilton. *Time Series Analysis*. Princeton University Press, Princeton, NJ, 1994. doi: 10.1515/9781400828896.

[6] Rob J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, Melbourne, Australia, 3rd edition, 2021. URL https://otexts.com/fpp3/.

[7] Eric Jan. *Econometrics with R – Section 8.2: Unit Root Testing*. 2023.

[8] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. M5 accuracy competition: Results, findings, and conclusions. *Forecasting and Strategy Unit, School of Electrical and Computer Engineering, National Technical University of Athens, Greece*, 2023. In press.

[9] Gianluca Mastrantonio. Dispense del corso di apprendimento statistico. Lezioni del corso, Politecnico di Torino, 2023.

[10] Meta Open Source (formerly Facebook). Prophet: Forecasting at scale (Documentazione Ufficiale). https://facebook.github.io/prophet/, 2024.

[11] Mohsen Mohseni. Time series forecasting: Continuous ranked probability score (crps). https://medium.com/@mohsenim/time-series-forecasting-continuous-ranked-probability-score-crps-ff5b8383d0e1, 2023.

[12] Ocean Insight / OptoSirius. *Principles of Optical Dissolved Oxygen Measurements*, —. URL https://www.optosirius.co.jp/OceanOptics/technical/principles-of-optical-dissolved-oxygen-measurements.pdf. Technical document, PDF.

[13] One Planet Summit. Brest commitments for the oceans. Technical report, One Planet Summit Secretariat, Brest, France, 2022. URL https://oneplanetsummit.fr/sites/default/files/2022-03/BREST-COMMITMENTS-FOR-THE-OCEANS.pdf.

[14] Skipper Seabold and Josef Perktold. statsmodels: Econometric and statistical modeling with python. Proceedings of the 9th Python in Science Conference, 2010. Software documentation of statsmodels (v0.14.4). Available online at https://www.statsmodels.org/stable/index.html.

[15] Smart Bay S. Teresa. Il progetto smart bay. https://smartbaysteresa.com/il-progetto-smart-bay/, 2024.

[16] Sean J Taylor and Benjamin Letham. Forecasting at scale. *PeerJ Preprints*, 5:e3190v2, 2017. doi: 10.7287/peerj.preprints.3190v2. URL https://peerj.com/preprints/3190v2/.

[17] R. F. Weiss. The solubility of nitrogen, oxygen and argon in water and seawater. *Deep-Sea Research*, 17, 1970.