2022

Progetto Industy Lab: Bacino Idrico

Mario Pedol, 830296
Nabil El Asri, 826040
26/08/2022

Contents

Abstract

L'obiettivo dell'elaborato consiste nel determinare con sette giorni di anticipo il livello d'acqua e la portata di uscita di un bacino idrico italiano al fine di limitare i rischi di alluvione, fornire risorse idriche per uso pubblico e generare energia idroelettrica.

Il risultato ottenuto sono due modelli di regressione in grado di effettuare le previsioni richieste, per i diversi target (livello d'acqua e portata d'uscita)

1. Introduzione

I laghi artificiali sono spesso alimentati da falde acquifere, la cui portata dipende principalmente dalle precipitazioni durante l'anno.

La società ha attivo dal 2003 un sistema sensoristico per il monitoraggio giornaliero del livello d'acqua e della portata d'uscita del bacino. L'obiettivo è quello di monitorare costantemente tali dati per efficientare i sistemi di gestione delle risorse idriche per uso pubblico, la produzione di energia idroelettrica e limitare i rischi di alluvione.

Per migliorare la precisione di tale sistema nel 2004, sono stati installati e attivati ulteriori sensori capaci di rilevare la quantità delle piogge nelle zone delle falde e la temperatura nella zona di una di queste. Con questo investimento si ritiene possibile sviluppare un sistema di modellistico Data Driven che sia sufficientemente accurato per prevedere con sette giorni di anticipo quale sarà il livello d'acqua e la portata d'uscita.

La prima previsione permetterebbe in momenti di potenziale criticità, eccessivo o scarso livello d'acqua, di intraprendere azioni manutentive in anticipo, come migliorare la tenuta degli argini qual ora ci fosse una potenziale rischio di esondazione, oppure pianificare azioni di riparazione fattibili solo con un livello molto basso di acqua.

Una previsione anticipata sulla portata di uscita invece permetterebbe di gestire meglio la produzione di energia idroelettrica capendo meglio quali sono i periodi dove la produzione potrebbe avere un improvviso abbassamento, per poter intervenire con strategie alternative a questi cali.

2. Dataset

Il dataset fornito per l'analisi fa riferimento a 8 variabili definite come:

1. Data: data della rilevazione;
2. Pioggia_Zona_1: rilevazione giornaliera della quantità piovuta nella zona 1;
3. Pioggia_Zona_2: rilevazione giornaliera della quantità piovuta nella zona 2;
4. Pioggia_Zona_3: rilevazione giornaliera della quantità piovuta nella zona 3;
5. Pioggia_Zona_4: rilevazione giornaliera della quantità piovuta nella zona 4;
6. Pioggia_Zona_5: rilevazione giornaliera della quantità piovuta nella zona 5;
7. Temperatura_Zona_5: rilevazione giornaliera della temperatura nella zona 5;

8. Livello_Acqua: livello d'acqua del bacino idrico rilevato all'interno della giornata (*target 1*);
9. Portata_uscita: Portata di uscita del bacino idrico rilevato all'interno della giornata (*target 2*).

I dati sono disponibili per i due target, dal 06/01/2003 al 29/06/2020, mentre per le restanti variabili dal 02/01/2004, questa mancanza è dovuta, come descritto nell'introduzione, all'installazione dei sensori con un anno di ritardo.

3. Analisi Esplorativa

Si indagano ora le variabili oggetto di studio per prendere maggiore confidenza con i dati e rilevare eventuali problematiche presenti in essi.

La prima cosa che salta all'occhio sono la presenza di valori mancanti per tutto l'anno 2003 per i rilevamenti della pioggia e della temperatura, questo fenomeno è dovuto al fatto che in tale anno i sensori non erano attivi o installati. Nonostante questo, è garantita sempre la continuità temporale della rilevazione giornaliera dei due target.

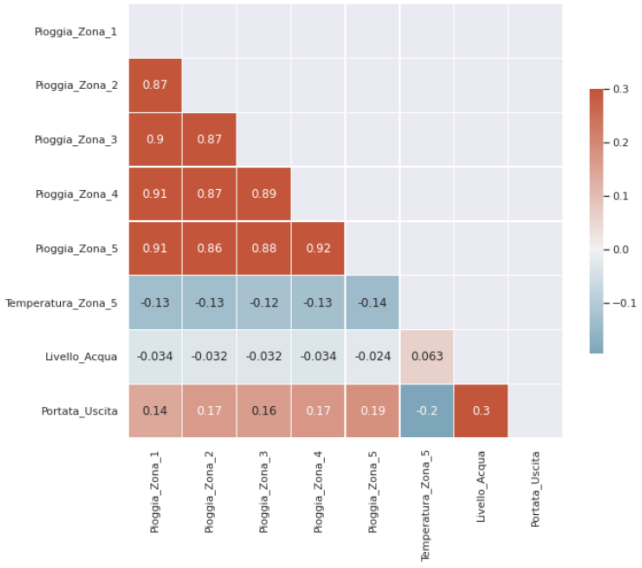

Figura 1: *Matrice di correlazione delle variabili*

Analizzando ora le correlazioni tra le variabili si nota una forte correlazione tra le zone di pioggia mentre i target risultano essere poco correlati con tutte le variabili, in particolare, il Livello di Acqua ha correlazioni tutte inferiori a 0,1; mentre la Portata d'uscita ha delle correlazioni più elevate, sopra lo 0,1; in particolare, con il Livello di Acqua ha una correlazione positiva dello 0.3.

Guardando l'andamento temporale dei due target nella Figura 2, si osserva come il Livello di Acqua abbia un trend costante e senza variazione, si ipotizza che la serie è stazionaria in media. La

Portata di Uscita, invece, ha un andamento più complesso, per la maggior parte degli anni, nei primi mesi si osservano picchi elevati, in particolare nel 2014 dove si raggiunge il massimo di 70 m^3/s.

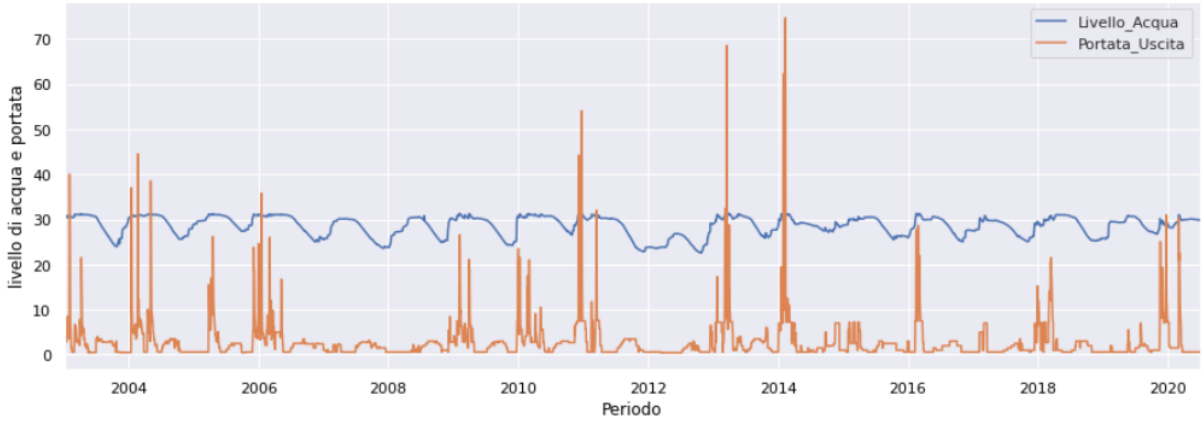

Figura 2: *Andamento temporale dei due target*

3.1 Analisi Statistica delle feature

Prima di processare i dati da passare ai modelli per l'addestramento, si è condotta un'analisi statistica sulle variabili target.

Considerando il *target 1* (Livello_Acqua) si testa se la serie è stazionaria in media con il test di Dickey Fuller, il p-value risultante è significativo a tutte le usuali soglie di significatività, si accetta quindi l'ipotesi nulla che la serie è stazionaria in media, dunque, si ha una conferma statistica di ciò che era stato precedentemente ipotizzato nel paragrafo 3 guardando la Figura 2.

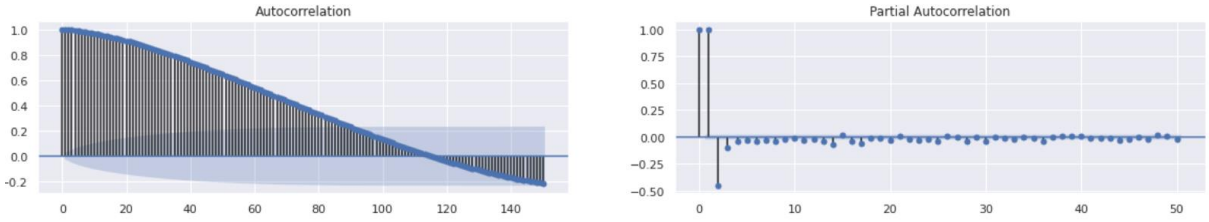

Figura 3: *Grafici di autocorrelazione (ACF) e autocorrelazione parziale (PACF) per la variabile Livello_Acqua.*

Analizzando i grafici di autocorrelazione e autocorrelazione parziale presenti nella Figura 3 si nota come l'ACF ha una correlazione significativamente alta nei lag iniziali e decresce molto lentamente (circa fino all'85 lag). Il PACF invece, presenta una correlazione significativa positiva alta nei primi 2 lag e negativa nel 3, poi tende ad annullarsi già dal 3 lag in poi.

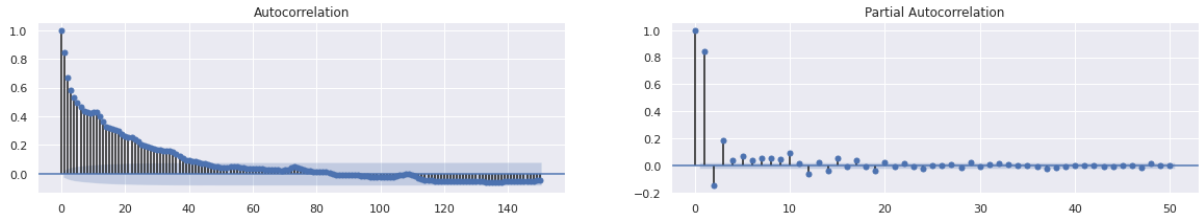

Figura 4: *Grafici di autocorrelazione (ACF) e autocorrelazione parziale (PACF) per la variabile Portata_Uscita*

Prendendo ora in considerazione il *target 2 (Portata_Uscita)* si applica di nuovo il test di Dickey Fuller per verificarne la stazionarietà. Il p-value risultante è significativo a tutte le usuali soglie di significatività; dunque, si accetta l'ipotesi nulla che la serie è stazionaria in media. A differenza del precedente target, risulta essere meno immediato avere una verifica visiva di stazionarietà dalla Figura 4; facendo particolarmente attenzione è possibile notare che i tanti picchi presenti sono effettivamente controbilanciati dai valori minimi di portata, anch'essi molto frequenti.

Dai grafici di autocorrelazione e autocorrelazione parziale presenti nella Figura 4 si osserva che l'ACF ha una correlazione significativamente alta nei lag iniziali e decresce lentamente fino al 40° lag. Il PACF ha una correlazione significativa positiva per i primi 2 lag, per poi tendere a zero dopo il 3° e 4° lag.

4. Processing

Una volta effettuate le analisi preliminari si pre-processano i dati per l'addestramento dei modelli seguendo 2 fasi:

1. Processamento del target per modelli Uni variati;
2. Processamento dell'intero Dataset per modelli multivariati.

Tali operazioni sono state svolte sia per predire il target 1 che il target 2, i dati dunque saranno pre-processati in 4 modi differenti.

4.1 Preprocessing Univariato

Considerando le analisi svolte fino a questo punto, non sono emerse criticità sulle variabili target, di conseguenza non è stata svolta nessuna manipolazione sui dati.

Il periodo che si è considerato per allenare il modello è tutto lo storico disponibile dal 01/06/2003 al 23/06/2019 mentre per il test dal 24/06/2019 al 22/06/2020. I dati per il *target 1* sono stati salvati nel come train_sarima.csv e test_sarima.csv nella cartella *"Data/Prepared"*.

In modo analogo, per la variabile Portata_Uscita, si è considerato tutto lo storico disponibile dal 01/06/2003 al 23/06/2019 per il training set, e lo storico dal 24/06/2019 al 22/06/2020 per il test. I dati sono stati salvati nella cartella *"Data/Prepared"*, rispettivamente come train_PU_SA e test_PU_SA.

4.2 Preprocessing Multivariato

Per quanto riguarda la preparazione dei dati per i modelli multivariati per il target 1 sono state svolte le seguenti operazioni:

- Eliminazione delle righe con valori mancanti;
- Creazione di 17 nuove variabili;
- Creazione della variabile target come shift della variabile target 1.

Poiché, come già visto al paragrafo 2, i valori mancanti delle rilevazioni di pioggia sono relativi a tutto l'anno 2003, essi sono rimossi, riducendo così lo storico dal 02/01/2004 al 30/06/2020.

Dopo aver svolto questo passaggio, si creano delle nuove feature che potrebbero essere potenzialmente utili ai modelli utilizzati per predire la variabile target. Queste vengono definite come:

1. Month: mese della rilevazione (1-12);
2. Year: anno della rilevazione;
3. Pioggia_mediana: mediana giornaliera delle piogge tra le 5 falde acquifere;
4. Piogga_std: varianza giornaliera delle piogge tra le 5 falde acquifere;
5. Winter: valore binario se inverno (1) o altro (0);
6. Summer: valore binario se estate (1) o altro (0);
7. Spring: valore binario se primavera (1) o altro (0);
8. Autumn: valore binario se autunno (1) o altro (0);
9. Livello_Acqua_lag(i): rilevazione del livello d'acqua del giorno i, con i=-1,...,-6;
10. Livello_Acqua__mediana: mediana del Livello d'acqua nei sette giorni precedenti;
11. Livello_Acqua__std: deviazione standard del Livello d'acqua nei sette giorni precedenti.

Come ultima operazione, poiché l'obiettivo è quello di predire il settimo giorno, si crea una nuova variabile target “shiftando” i valori della variabile obiettivo *(y=Livello_Acqua)* indietro di 7 giorni (14 dal dataset di partenza), così che ad ogni valore di *y* al giorno *i-esimo* sono associate le variabili dal i al giorno *i-7*, più le rilevazioni del target nei 6 giorni precedenti.

Tale operazione porta a una riduzione dello storico disponibile di 7 giorni, avendo così un periodo risultante finale che va dal 02/04/2004 al 23/06/2020.

I dati risultanti poi sono splittati in train e test set, dove il train va dal 01/04/2004 al 23/06/2019, mentre il test ha lo stesso periodo del test set univariato. Questi sono salvati come *train_SL.csv* e *test_SL.csv* nella cartella *“Data/Prepared”*.

Per il secondo target la preparazione dei dati è stata diversificata al fine di ottenere un set più congruo alla modellazione della Portata_Uscita. Le uniche operazioni in comune sono state l’eliminazione delle righe con valori mancanti e la creazione della nuova variabile target prendendo lo shift di 7 giorni della Portata_Uscita, in linea con l’obiettivo della previsione anticipata di una settimana. Dunque, la vera differenza riguarda la creazione delle variabili aggiuntive, che possono essere sintetizzate nel modo seguente:

1. Winter: valore binario, se inverno (1) altrimenti (0);
2. Summer: valore binario, se estate (1) altrimenti (0);
3. Spring: valore binario, se primavera (1) altrimenti (0);
4. Autumn: valore binario, se autunno (1) altrimenti (0);
5. Year: anno della rilevazione;
6. Month: mese della rilevazione (1-12);
7. Day_in_year: giorno nell’anno (1-365);
8. Week_in_year: settimana nell’anno (1-52);
9. Temperatura_Trend: componente trend-ciclo della variabile Temperatura_Zona_5;
10. Temperatura_Season: componente stagionale della variabile Temperatura_Zona_5:
11. Temperatura_Resid: componente irregolare della variabile Temperatura_Zona_5;
12. Pioggia_Zona_j_diff_{i}: differenza tra i valori di Pioggia_Zona_j al tempo t e t-i con i che va da 1 a 6, j da 1 a 5 (tutte le zone);
13. Livello_Acqua_diff_{i}: differenza tra i valori di Livello_Acqua al tempo t e t-i con i che va da 1 a 6;
14. Portata_Uscita_diff_{i}: differenza tra i valori di Portata_Uscita al tempo t e t-i con i che va da 1 a 6;

15. Temperatura_Trend_Shifted: valore di Temperatura_Trend del giorno precedente;
16. Temperatura_Season_Shifted: valore di Temperatura_Season del giorno precedente;
17. Temperatura_Resid_Shifted: valore di Temperatura_Resid del giorno precedente;
18. Livello_Acqua_Shifted: valore di Livello_Acqua di 30 giorni prima;
19. Portata_Acqua_Shifted: valore di Portata_Acqua di 30 giorni prima.

Dopo l'aggiunta di queste variabili, il periodo risultante che si è ottenuto parte dal giorno 03/07/2004 e termina al giorno 30/12/2019. Lo split è stato fatto prendendo l'80% dei dati come train set e il restante 20% come test set. Questi sono salvati come train_PU_SL e test_PU_SL nella cartella *"Data/Prepared"*.

5. Modelli

Per affrontare in maniera adeguata il task dell'elaborato si è deciso di vagliare 2 strade, la prima sfrutta le variabili target in maniera univariata considerando l'intero periodo, la seconda, tiene in considerazione tutte le variabili ottenute durante la fase di preprocessing multivariato.

I modelli presi in considerazione per le 2 differenti strade sono:

1. Modelli Univariati:
    - SARIMAX: Con tale acronimo si intende,Seasonal Auto-Regressive Integrated Moving Average with eXogenous factors, una particolare tipologia di modelli atti ad indagare serie storiche che presentano caratteristiche particolari. Presenta 7 parametri che possono essere descritti come (p,d,q) (P, D,Q,s), dove p indica il numero dei termini autoregressivi, d è il numero di differenziazioni non stagionali per la stazionarietà, q indica il numero di errori di previsione ritardati nell'equazione di previsione. P,D,Q sono le componenti corrispettive stagionali delle precedenti, in aggiunta il termine indica il numero di periodi necessari prima che la tendenza ricompaia.
2. Modelli Multivariati: I modelli multivariati scelti sono due Ensemble, ossia la combinazione di modelli semplici così che il modello finale completo diventa un perditore più forte, entrambi hanno come modello di base l'albero di decisione usato come regressore. La scelta è ricaduta su questi in quanto sono in grado di generalizzare al meglio i dati ed evitare problemi di overfitting, inoltre, come appena detto sono perditori che risultano più robusti rispetto ai modelli classici:

- XGBoostRegressor: Il "boosting" nell'apprendimento automatico è un modo per combinare più modelli semplici in un unico modello. Il termine "gradiente" in "gradient boosting" deriva dal fatto che l'algoritmo utilizza la discesa del gradiente per minimizzare la funzione di perdita e raggiungere performance più elevate.
- RandomForestRegressor: è un modello che adatta una serie di alberi decisionali su vari sotto campioni del data set e utilizza la media dei risultati per migliorare l'accuratezza e controllare l'overfitting.

5.1 Training Target 1

Per addestrare i dati con il modello univariato SARIMAX si è deciso di sfruttare la libreria python "pmdarima", essa permette di avvalersi del metodo auto_arima() che fitta i valori del dataset di train al fine di trovare i migliori valori per i parametri del modello confrontando i valori dell'AIC e scegliendo quello con il valore più basso.

L'ordine dei parametri ottimo ottenuto è (2,1,3) (0,0,1,24), dove il valore di differenziazione 1 è dovuto al fatto che la serie originale ha un trend medio costante.

facendo una rapida diagnostica del modello si nota che i parametri sono tutti significativi e i residui sono approssimativamente distribuiti come una normale. Il modello viene dunque salvato nella cartella *"Models/Tentativi"* come model_arima.pkl.

Considerando ora i modelli multivariati, Random Forest e XGBoost, gli iper-parametri per ciascun modello sono stati scelti definendo una griglia di valori per ciascun iper-parametro e utilizzando una tecnica di cross validation per le serie temporali, ossia facendo 10 split in modo tale da non mescolare l'ordine tra train e validation, assicurandoci che i dati di train vengono prima dei dati di validation ad ogni step. La combinazione dei parametri finali viene poi scelta come il modello migliore su tutti i k-fold, dove il modello migliore è il modello con lo score più prossimo a 0 in termini di *neg_mean_squared_error,* ossia l'errore quadratico medio negativo definito come:

$$-MSE = -\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Dove:

N, numero delle osservazioni;

y_i, sono i valori reali osservati al tempo i-esimo;

$\hat{y}_i$, sono i valori predetti dal modello al tempo i-esimo.

Ciò è stato possibile implementarlo grazie al metodo GridSearchCV della libreria sklearn.

Una volta trovati i migliori parametri si è voluto applicare un algoritmo per scegliere il numero di variabili ottimali per ogni modello. In particolare, è stata scelta la recursive feature selection con cross validation (stessa tecnica di validazione per la selezione degli iper-parametri). Questa tecnica prevede di fittare k=10 modelli eliminando ricorsivamente una variabile per volta a seconda della sua importanza.

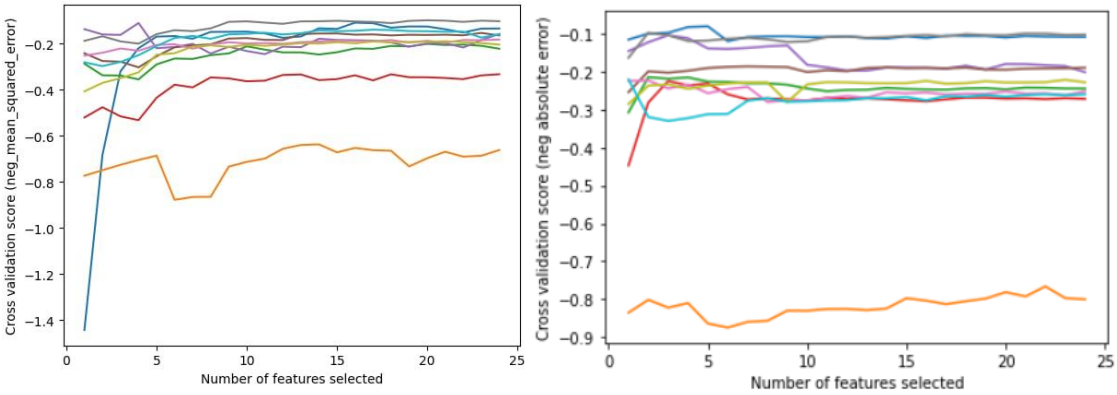

Figura 5: *Andamento degli score (train) all'aumentare delle variabili, XGBoost a sinistra, RandomForest a destra.*

Come si nota dai risultati della Figura 5, i valori degli score a seconda del numero di variabili per ogni fold aumentano all'aumentare delle variabili per entrambi i modelli, lo score sembra stabilizzarsi sopra la 13 variabile per XGBoost e la 6 per il RandomForest.

Come suggerito dall'algoritmo si prendono in considerazione le variabili che massimizzano lo score in media tra tutti i 10 gli splits, esse sono:

- Random Forest: si selezionano solo le seguenti 5 variabili: "Pioggia_Zona_2", "Temperatura_Zona_5", "Portata_Uscita", "summer", "Livello_Acqua_lag0";
- XGBoost: anche se l'algoritmo restituisce una selezione di sole 16 variabili, si considerano tutte poiché la variazione dello score è minima.

Una volta individuati i due modelli migliori e addestrati, essi sono stati salvati nella cartella Models come XGB_model_LA e RF_model_LA_featsel. Inoltre, per essere sicuri dell'efficacia della feature selection del Random Forest, e evitare un possibile overfitting, visto che sembra non variare molto lo score all'aumentare delle feature, si salva anche il Random Forest addestrato con tutte le variabili come RF_model_LA.

5.2 Training Target 2

Per quanto riguarda l'addestramento con il modello univariato SARIMAX, si è adottata una strategia speculare a quella utilizzata con il target 1, sfruttando la libreria python "pmdarima" per utilizzare auto_arima() ed individuando così i migliori valori per i parametri del modello.

L'ordine dei parametri che minimizzano l'AIC è (2, 0, 2) (0, 0, 0, 12), escludendo quindi tutte le componenti stagionali.

I parametri risultano essere significativi e approssimativamente distribuiti come una normale, ipotesi confermata dal test di Durbin-Watson. Esso è stato salvato nella cartella *"Models/Tentativi"* come model_arima_PU.pkl.

Per la parte dei modelli multivariati, è stata utilizzata anche in questo caso la griglia dei valori per ciascun iper-parametro in modo tale da ottimizzare il modello fin dalla sua configurazione; inoltre, si è adottata, come fatto precedentemente, la tecnica di cross validation per serie temporali, al fine di garantire l'ordine temporale dei dati tra train e validation. La scelta finale dei parametri viene fatta valutando lo score dell'errore quadratico medio negativo delle diverse configurazioni su tutti i k-fold.

Dato il numero elevato di variabili inserite nel secondo pre-processamento dei dati, è stata implementata la RandomizedSearchCV, che a differenza della GridSearchCV, non va a ricercare tutte le combinazioni possibili ma conserva man mano i valori che danno risultati migliori e li combina tra di loro. Sempre in ottica di ridurre il tempo di esecuzione, il cross validation per l'XGBoost è stato eseguito su 10 split mentre per il Random Forest su 3, poiché quest'ultimo richiede maggior tempo rispetto al primo.

infine, una volta individuata la configurazione migliore dei parametri, si è applicata la Recursive Feature Selection con cross validation per valutare l'importanza delle singole variabili.

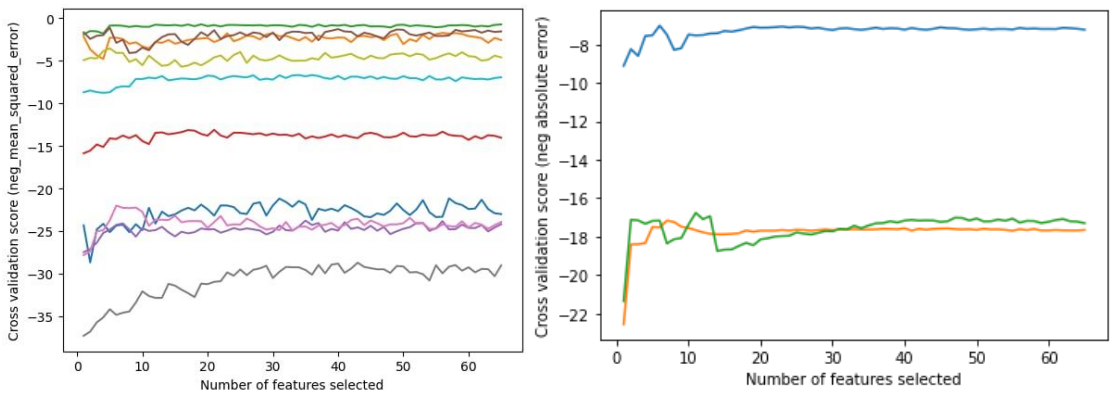

Figura 6: *Andamento degli score (train) all'aumentare delle variabili, XGBoost a sinistra, RandomForest a destra, target2.*

Come visto in precedenza per il target 1, anche in questo caso i valori degli score aumentano all'aumentare delle variabili per tutti i fold esaminati.

L'algoritmo suggerisce i seguenti modelli ottimali:

- XGBoost con 59 variabili su 65;
- Random Forest con 6 variabili, ovvero "Livello_Acqua", "day_in_year", "Temperatura_Trend", "Portata_Uscita_diff_6", "Temperatura_Season_shifted" e "Livello_Acqua_shifted".

Per scrupolo e per fare delle valutazioni più approfondite, oltre ai modelli ottimali XGB_model_featsel_PU e RF_model_featsel_PU, sono stati salvati anche i restanti due che comprendono tutte le variabili, rispettivamente XGB_model_PU, RF_model_PU nella cartella *"Models"*.

6. Test e Selezione del modello migliore

Una volta addestrati i modelli si testa la loro efficacia sul test set definito in fase di preprocessing, scegliendo quello con il MSE minore.

Poiché il modello statistico SARIMAX si basa sullo storico precedente e prevede il tempo futuro, tendendo alla media dopo pochi step avanti, è stato necessario valutarlo mediante una previsione rolling, dove iterativamente, al tempo t si predice il valore t+7 e al tempo t+1, t+8 e così via, aggiungendo allo storico ad ogni iterazione il valore osservato successivo. Questa strategia permette di valutare in maniera coerente tutti i modelli addestrati.

6.1 Target 1

Si analizzano ora i risultati per il target 1, riportati nella Tabella 1.

Modelli	Score (mse)
SARIMAX	0.222
RF_model_LA_featsel	0.215
XGB_model_LA	0.157
RF_model_LA	0.181

Tabella 1: *score per modello (mse) per il target 1.*

Tutti i modelli si comportano abbastanza bene, l'errore è piuttosto basso, il modello univariato ha lo score peggiore, questo può significare che l'utilizzo di più variabili aiuta maggiormente a predire il target.

Tra i modelli multivariati si nota come la feature selection per il Random Forest abbia leggermente peggiorato le performance, si può pensare a un leggero overfitting in fase di train, mentre l'XGBoost presenta lo score più basso tra tutti i modelli.

Dunque, il modello migliore selezionato è XGB_model_LA, ossia il modello XGBoost con tutte le variabili. In seguito, si riporta il grafico con le previsioni sul test set.

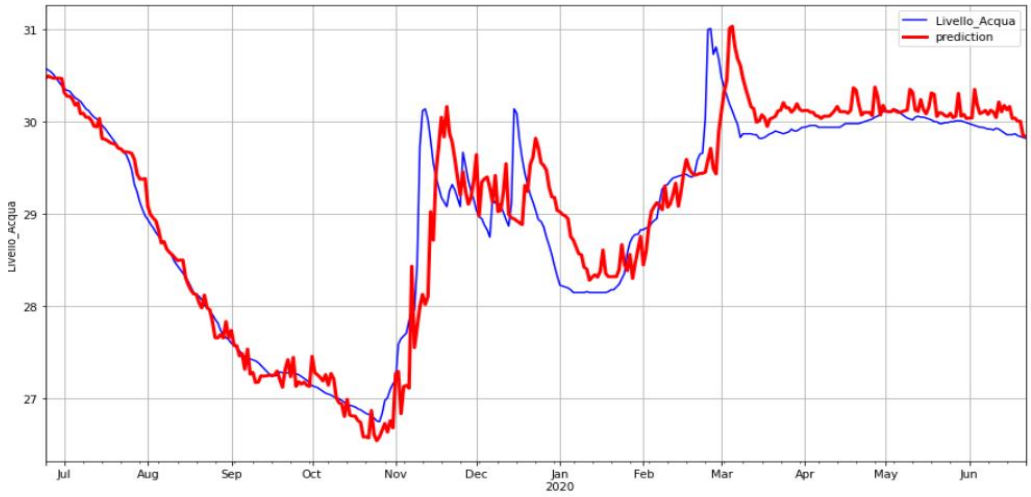

Figura 7: *Valori predetti dal modello XGB_model_LA vs Valori osservati per la variabile Livello_Acqua.*

6.2 target 2

Si riportano ora i risultati per il secondo target nella tabella 2.

Modelli	Score(mse)
model_arima_PU	24,33
XGB_model_PU	7,73
XGB_model_featsel_PU	8,1
RF_model_PU	9,37
RF_model_featsel_PU	9,09

Tabella 2: *score per modello (mse) per il target 2.*

Si nota subito una sostanziale differenza in termini di MSE tra il modello univariato e i restanti modelli multivariati. Infatti, lo score risulta essere quasi 3 volte superiore all'XGBoost e al Random Forest, segno che le variabili sono estremamente utili per prevedere il target Portata_Acqua.

Il modello che si comporta meglio è l'XGBoost con tutte le variabili, sebbene la differenza sia davvero minima con il modello con meno variabili. Si osserva inoltre come gli score siano più alti rispetto a quelli individuati per il target precedente; questo perché, come è stato già osservato nella Figura 2, l'andamento della Portata_Acqua nel tempo è più complesso, con dei picchi che si verificano con cadenze e valori diversi. Questo ha fatto sì che anche il modello migliore produca delle previsioni non sempre ottimali, soprattutto all'altezza dei picchi positivi. Si riporta il grafico del modello XGB_model_featsel_PU con le previsioni sul test set.

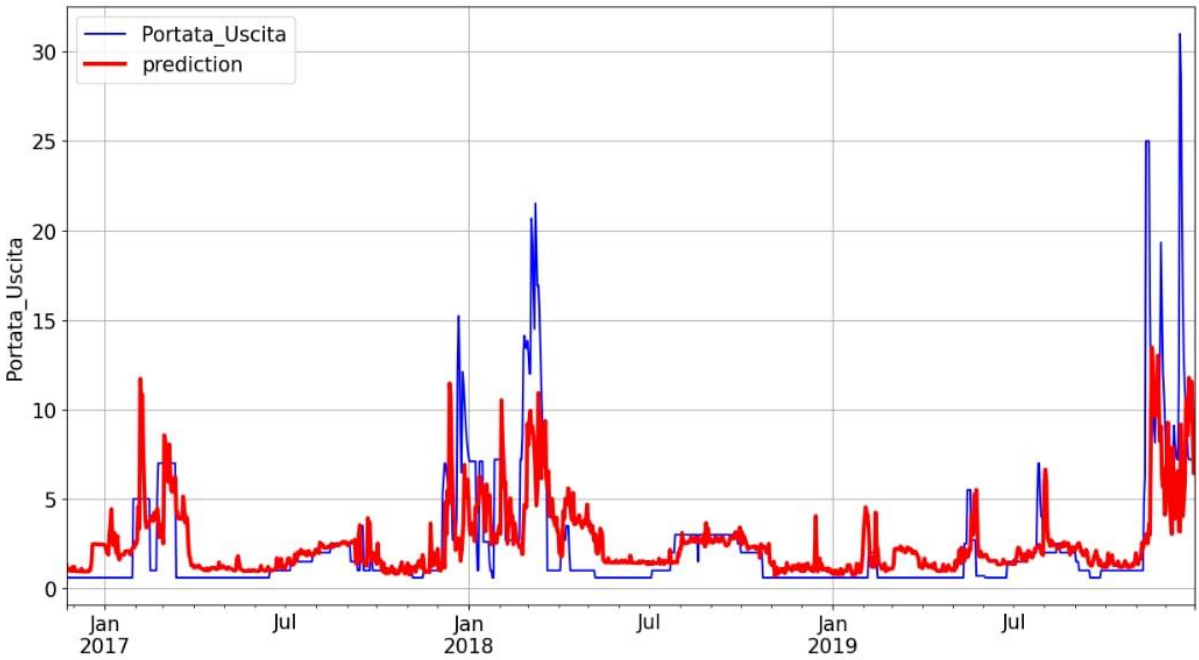

Figura 8: *Valori predetti dal modello XGB_model_featse_PU vs Valori osservati per la variabile Portata_Uscita*.

7. Analisi dei modelli migliori

Una volta selezionati i modelli è utile comprendere come le variabili impattino su di essi.

In questo paragrafo, si vuole dunque indagare l'importanza delle variabili al fine di dare una spiegazione più completa dei modelli che andranno in produzione.

Per analizzare tale importanza è stato scelto come metodo quello dello *Shap Value* della libreria python "shap". Tree SHAP è un algoritmo per calcolare i valori esatti di SHAP per i modelli basati sugli alberi decisionali. SHAP (SHapley Additive exPlanation) è un approccio basato sulla teoria dei giochi per spiegare l'output di qualsiasi modello di apprendimento automatico. L'obiettivo è spiegare la previsione di qualsiasi istanza x_i come somma dei contributi dei valori delle singole variabili.

7.1 Target 1

Nella Figura 9 sono riportati i risultati dell'importanza delle variabili sui dati di test.

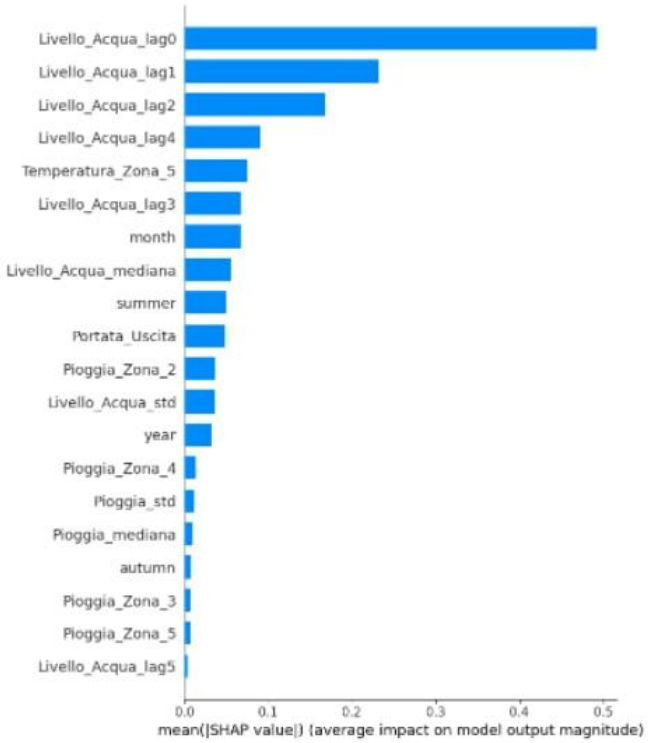

Figura 9: *Importanza delle variabili SHAP per il target Livello_Acqua, misurata come valori assoluti medi di Shapley.*

Si osserva, come ci si potrebbe aspettare, che le variabili che hanno un impatto maggiore per l'apprendimento del valore da predire sono il livello dell'acqua al tempo t (giorno in cui mi trovo) e al tempo t-1. Le variabili invece che risultano avere un impatto minore sono le piogge nelle zone 3,5 e il valore del livello di acqua al tempo t-5, ultimo valore utile alla previsione.

Si nota anche come le rilevazioni della temperatura nella zona 5 aiutano molto il modello rispetto alle piogge.

7.2 Target 2

Si analizzano le variabili utili per prevedere i valori di portata dell'acqua, ordinate per importanza e riportate (prime 20) nella figura 10.

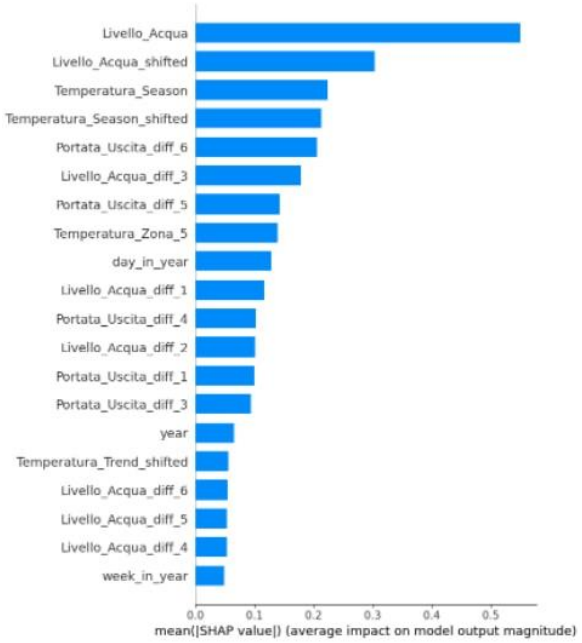

Figura 10: *Importanza delle variabili SHAP per il target Portata_Acqua, misurata come valori assoluti medi di Shapley.*

Le variabili che risultano essere maggiormente utili allo scopo sono quelle che rilevano il livello dell'acqua (ovvero target 1) e la componente stagionale della temperatura (zona 5). Le variabili costruite mediante la differenza dei valori nel tempo t hanno funzionato se si considerano gli attributi target Livello_Acqua e Portata_Uscita; non si può dire lo stesso per le variabili di pioggia. D'altronde le informazioni sul passato del target di riferimento, unite a quelle del target 1 e alla temperatura, sono sufficienti per modellare l'andamento (seppur con imprecisione).

8. Conclusioni

Il modello per la previsione del Livello dell'acqua sembra essere abbastanza soddisfacente per la previsione a 7 giorni, per predire un nuovo valore, saranno necessari sempre gli ultimi sette giorni in modo da poter ricostruire il dato di input in maniera corretta.

La Figura 11 mostra la previsione del modello a 7 giorni dall'ultima data disponibile per i dati forniti. Il valore predetto sembra essere sensato seguendo il trend decrescente degli ultimi mesi.

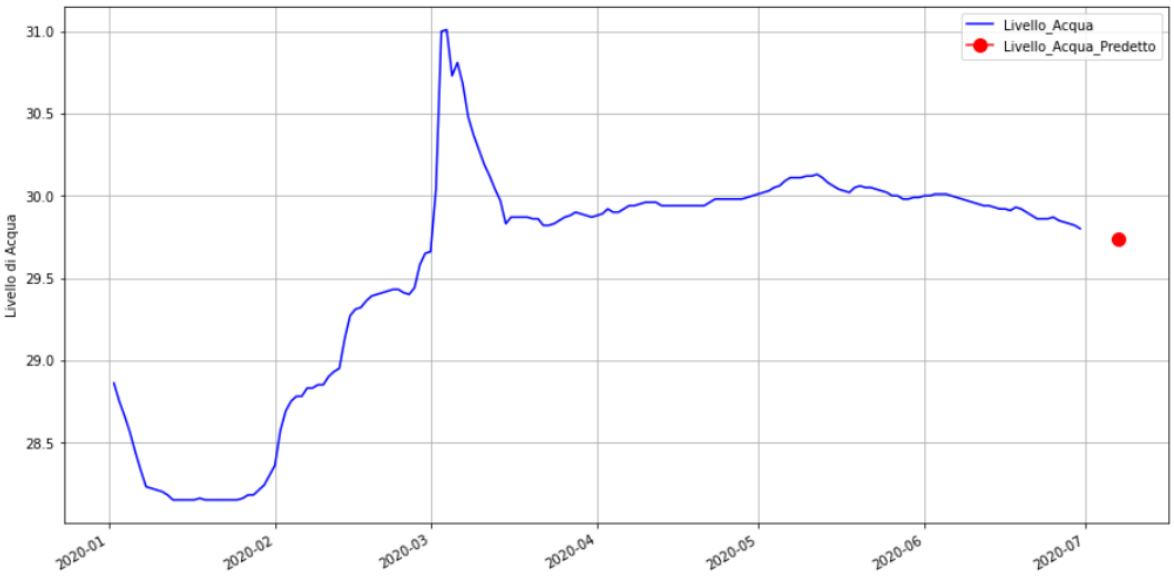

Figura 11: *Previsione a distanza di una settimana del Livello_Acqua con il modello XGB_model_LA.*

Per quanto riguarda il modello che prevede la portata d'uscita, considerando la complessità della serie, ci si può ritenere soddisfatti se si guarda all'andamento generale. In particolare, la crescita e la decrescita della portata vengono previste senza particolari problemi. Il discorso cambia invece se si considerano i valori effettivi dei picchi, specialmente quelli positivi. Il modello prevede con un errore medio pari a 7.73, restituendo spesso valori inferiori a quelli reali e quindi sottostimando i picchi di portata.

Durante il processamento dei dati è stata valutata una strada alternativa rispetto a quelle descritte nell'elaborato per cercare di gestire questa problematica. Si è ipotizzato che la causa dei valori sottostimati sia da attribuire ai tanti valori sulla soglia dello zero che costituiscono la serie. Dunque, è stato fatto un preprocessing orientato a registrare valori massimi e minimi per poi utilizzarli nella normalizzazione della serie. Tuttavia, i risultati sono stati inferiori alle attese e si è scelto di continuare con il pre-processamento originale. Un possibile sviluppo potrebbe essere quello di analizzare cosa sia andato bene e cosa sia andato male nella normalizzazione, provando diverse formule presenti in letteratura e registrando i possibili miglioramenti nei valori dei picchi.

Tornando al modello selezionato, si riporta la previsione a 7 giorni di distanza dall'ultima data disponibile nei dati forniti. Il modello prevede un picco notevole nel settimo giorno di luglio, confrontando i dati del medesimo periodo negli anni precedenti si osserva che il mese è solitamente caratterizzato da un picco che si contrappone ai valori vicini allo zero di fine giugno/inizio luglio.

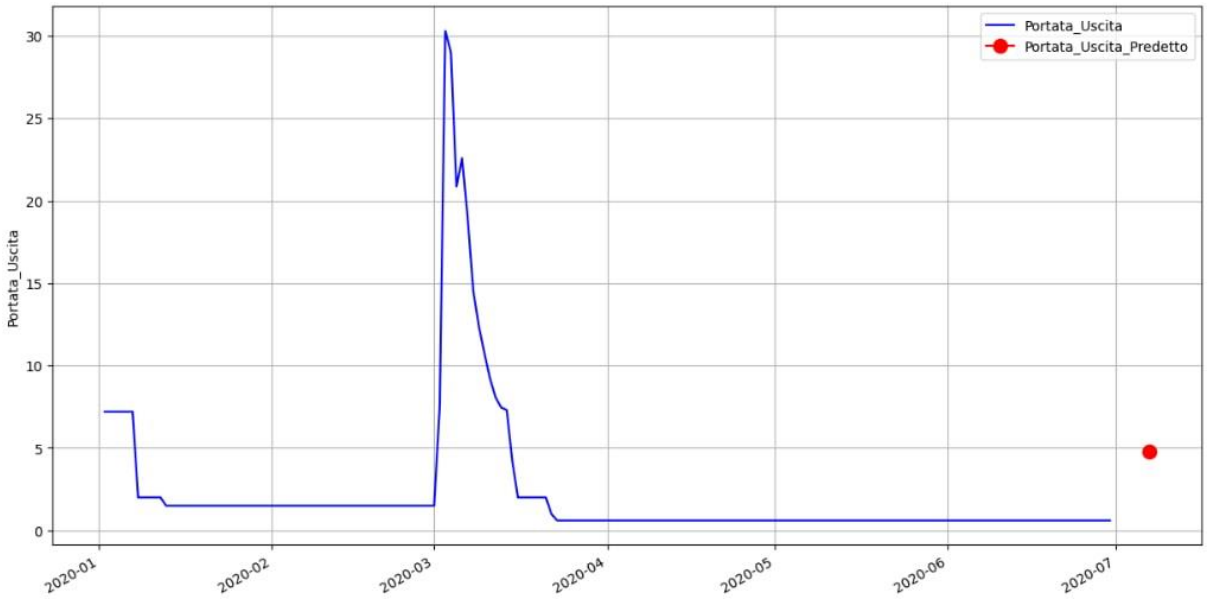

Figura 12: *Previsione a distanza di una settimana della Portata_Uscita con il modello XGB_model_featsel_PU.*

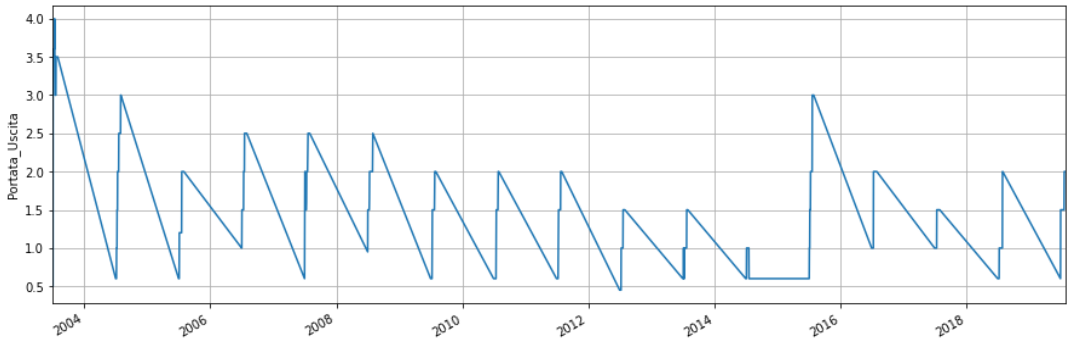

Figura 13: *Valori osservati per la variabile Portata_Uscita nei mesi di luglio, dal 2003 al 2019.*

9. Riferimenti

[1] Presentazione progetto Bacino Idrico 27/04

[2] https://christophm.github.io/interpretable-ml-book/shap.html