Università degli Studi di Milano Bicocca

Scuola di Scienze

Dipartimento di Informatica, Sistemistica e Comunicazione

Corso di laurea magistrale in Data Science

Predizione del livello d'acqua e della portata d'uscita di un bacino idrico

Luglio 2022

Merlo Riccardo - 829805 - r.merlo2@campus.unimib.it
Riva Leonardo - 830647 - l.riva37@campus.unimib.it
Stoffa Giacomo - 830159 - g.stoffa1@campus.unimib.it

Abstract

Il progetto si sviluppa nell'ambito di industry 4.0. L'obiettivo consiste nella previsione del livello d'acqua e di portata di uscita di un bacino idrico italiano. Dopo un'analisi esplorativa, sono stati preparati i dati in modo tale da poter allenare dei modelli di regressione: il focus principale è dato dai vincoli del problema, ossia gli offset (tempo che intercorre tra l'ultimo dato e la previsione) e i prev (quanti dati passati dare in input ai modelli). Successivamente, sono stati sviluppati e testati i seguenti modelli: SVR, KNN, reti neurali (FNN e LSTM). Individuati i migliori modelli per ciascuna variabile target (SVR per il livello e FNN per la portata), sono state effettuate le previsioni sul test set. Sulla variabile target livello, la SVR ha ottenuto, in un range di offset tra 1 e 28, dei MAE tra 0.868 e 0.937; per la variabile portata, invece, la FNN ha performato tra 1.340 e 2.477.

Indice

1. Introduzione

Background

Il progetto si inserisce nell'ambito di Industry 4.0, in particolare nel contesto di gestione dell'acqua in un bacino idrico. Il bacino in questione è stato creato appositamente per limitare i rischi di alluvione nella zona; viene alimentato da cinque falde acquifere, le quali portano acqua da cinque zone geografiche diverse, emettendo acqua in un'unica uscita. Fornisce così risorse idriche per uso potabile o per produrre energia elettrica con l'obiettivo di usare questa risorsa in modo sostenibile e garantendone qualità e integrità.

Contestualizzazione industry 4.0

Con il termine industry 4.0 si descrivono tutti i cambiamenti e miglioramenti, in ambito industriale, dovuti all'introduzione e al sempre maggiore utilizzo di sistemi interconnessi e di automazione, al fine di migliorare le condizioni e la qualità del lavoro. Nello specifico dello use-case in esame, abbiamo diversi tipi di sensori che generano dati con cadenza giornaliera. Questi dati hanno come obiettivo il monitoraggio del livello d'acqua presente nel bacino idrico e della sua portata di uscita. L'importanza dei sensori e l'arte dello studio dei dati sono concetti dominanti ed essenziali nell'industria 4.0; infatti, questo specifico tipo di analisi è conosciuto, all'interno di quest'ambito, come problema di predictive maintenance con sensori intelligenti IoT. In casi d'uso dove è il dato trattato è privato, e quindi la sua consistenza deve avvenire in locale, parliamo di processo di integrazione industriale, nel quale tutte le operazioni partendo dalla rilevazione fino ad arrivare alla costruzione dei report con annesse previsioni avvengono in un ambiente chiuso e quindi disconnesso da internet.

Obiettivi

Il progetto ha lo scopo di ottimizzare le procedure di gestione dell'acqua, anticipando eventuali carenze o segnalando possibili eccessi, utilizzando un sistema di previsione basato su modelli di regressione. Le predizioni del livello d'acqua e della portata di uscita verranno inizialmente effettuate con una settimana di anticipo; successivamente, verrà studiato come cambieranno le performance dei modelli in caso di rilassamento (1, 3 o 5 giorni) o rafforzamento di questo vincolo (14, 21 o 28 giorni).

2. Esplorazione

Dataset

Il dataset ha la seguente struttura:

Nome del campo	Tipo	Descrizione
Data	String	Identificativo del giorno in cui sono state fatte le misurazioni (es. “13/01/2003”)
Pioggia_Zona_X	Float	Quantità di acqua caduta nel giorno nell’area X, espressa in millimetri (5 feature)
Temperatura_Zona_X	Float	Temperatura massima rilevata nel giorno nell’area X, espressa in gradi centigradi (1 feature)
Livello_Acqua	Float	Livello massimo d’acqua del bacino idrico rilevato nel giorno, espresso in metri rispetto alla massima profondità del bacino (Target #1)
Portata_Uscita	Float	Massimo flusso d’acqua in uscita dal bacino idrico rilevato nel giorno, espresso in metri cubi al secondo (Target #2)

Il dataset è composto da 6386 misurazioni, dal 6 Gennaio 2003 al 30 giugno 2020.

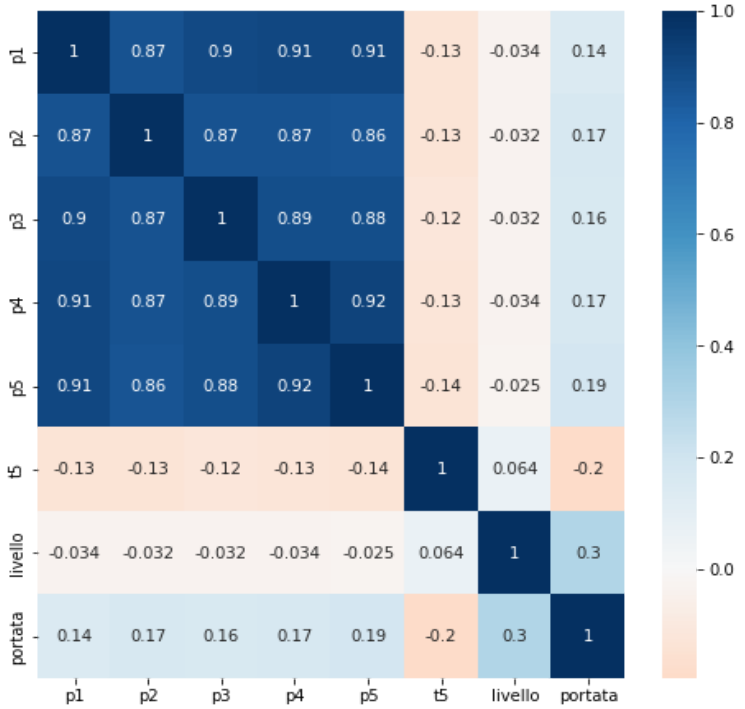
data	p1	p2	p3	p4	p5	t5	livello	portata
2003-01-06	NaN	NaN	NaN	NaN	NaN	NaN	30.70	9.6
2003-01-07	NaN	NaN	NaN	NaN	NaN	NaN	30.70	9.5
2003-01-08	NaN	NaN	NaN	NaN	NaN	NaN	30.67	9.5
2003-01-09	NaN	NaN	NaN	NaN	NaN	NaN	30.66	7.2
2003-01-10	NaN	NaN	NaN	NaN	NaN	NaN	30.64	6.2
...	...	...	...	...	...	...	...	...
2020-06-26	0.0	0.0	0.0	0.0	0.0	22.50	29.85	0.6
2020-06-27	0.0	0.0	0.0	0.0	0.0	23.40	29.84	0.6
2020-06-28	0.0	0.0	0.0	0.0	0.0	21.50	29.83	0.6
2020-06-29	0.0	0.0	0.0	0.0	0.0	23.20	29.82	0.6
2020-06-30	0.0	0.0	0.0	0.0	0.0	22.75	29.80	0.6

6386 rows × 8 columns

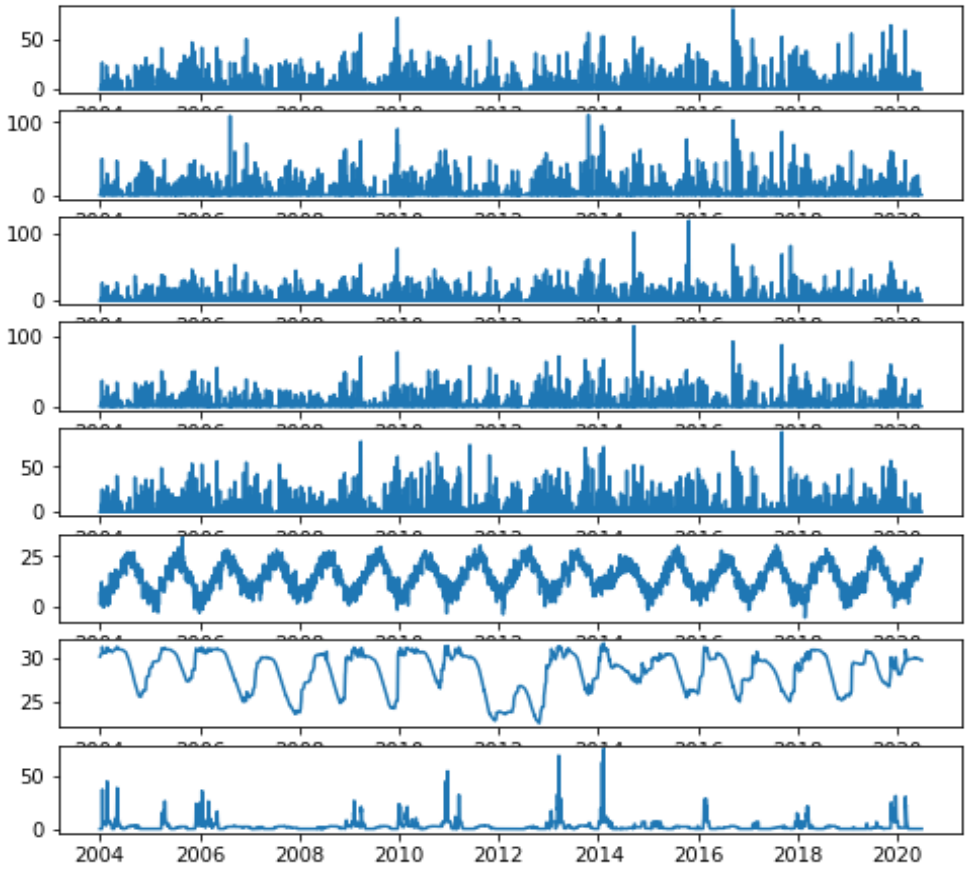
Per tutto il primo anno mancano i valori di ogni sensore (ad esclusione di livello e portata). I valori nulli corrispondono a misurazioni non effettuate per via di malfunzionamenti dei sensori o perché non ancora installati nella data corrispondente. Non risulta presente anche il valore di temperatura per il giorno 1 gennaio 2004.

Feature	% missing values
p1	5.64%
p2	5.64%
p3	5.64%
p4	5.64%
p5	5.64%
t5	5.65%
livello	0%
portata	0%

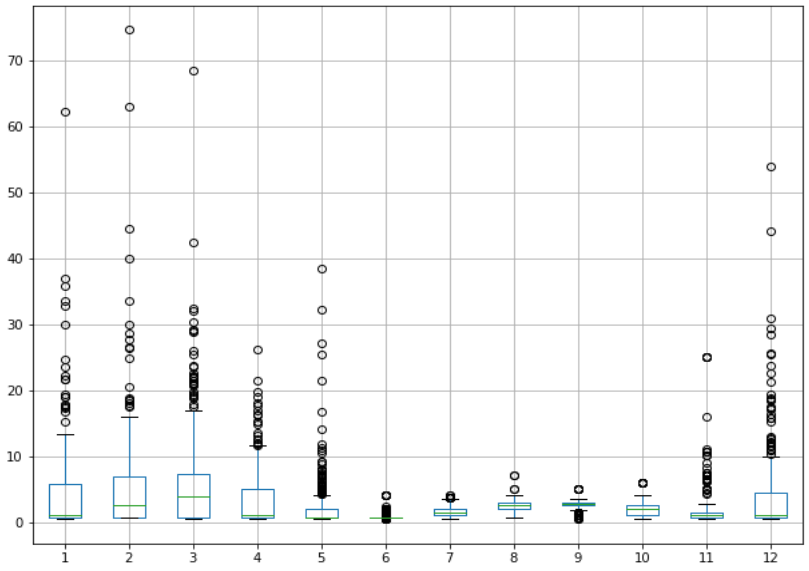
La matrice di correlazione mostra i coefficienti di correlazione tra le coppie di variabili, permettendo di valutare il grado di interdipendenza tra le varie feature. Risulta evidente la correlazione tra i parametri rilevati dai sensori di pioggia nelle 5 zone dei 5 affluenti, mentre rimangono sostanzialmente incorrelati con i valori di “Temperatura_zona_5”, livello d’acqua e portata di uscita del bacino idrico principale.



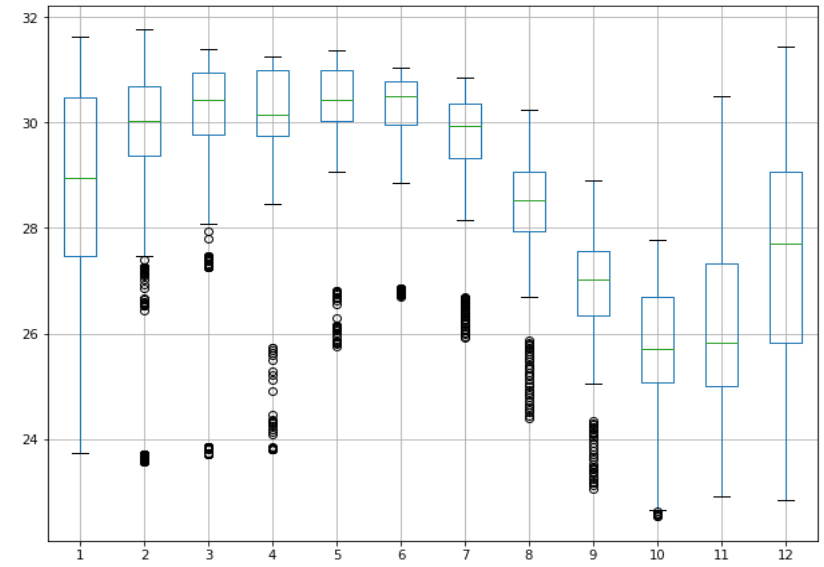
Visualizzando le distribuzioni di ciascuna feature, si può notare una certa stagionalità nell'andamento dei valori di livello, dovuta ovviamente al periodo dell'anno.



Tramite i seguenti boxplot si può avere una panoramica della variabilità dei valori di target (livello e portata) per ogni mese. Si può vedere, ad esempio, come nel corso di un anno il livello d'acqua del bacino idrico principale tenda a scendere verso giugno e risalire verso novembre.

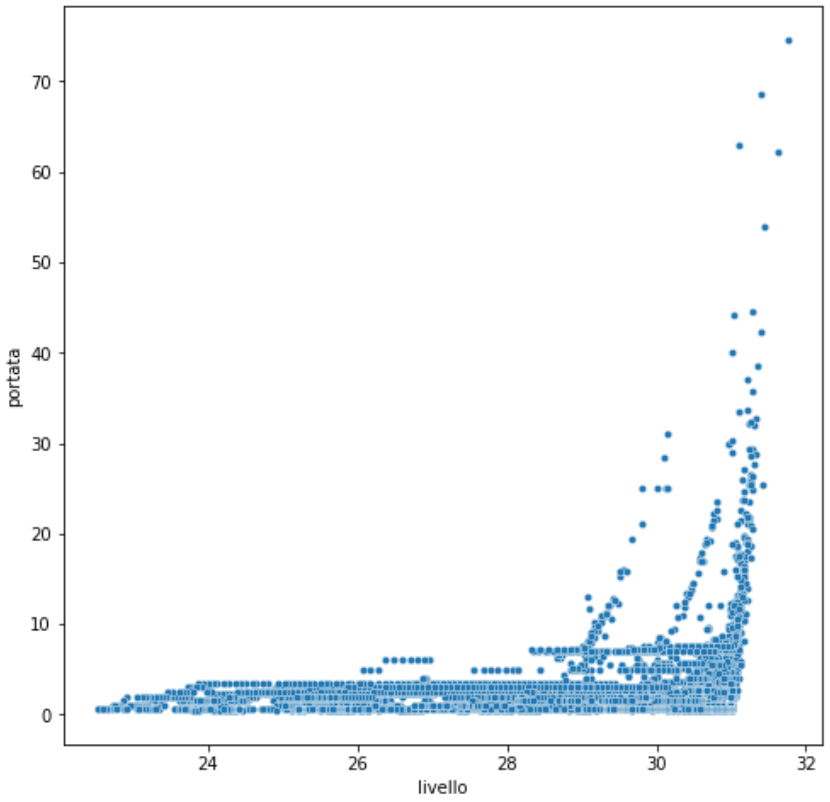


*Boxplot dei valori della variabile target **livello** raggruppati per mese.*



Boxplot dei valori della variabile target ***portata*** *raggruppati per mese.*

Inoltre, nello scatter plot in figura, è possibile notare come all'aumentare del valore del livello aumentano anche i valori "outlier" di portata.



ScatterPlot dei valori delle variabili target di livello e portata

3. Preparazione dei dati

Preprocessing

Per quanto riguarda i **missing values**, i dati del primo anno sono stati scartati, poiché mancavano tutti i dati utili alla previsione. Per il singolo missing value della temperatura del 1 gennaio 2004, è stato stimato arbitrariamente, osservando i valori successivi.

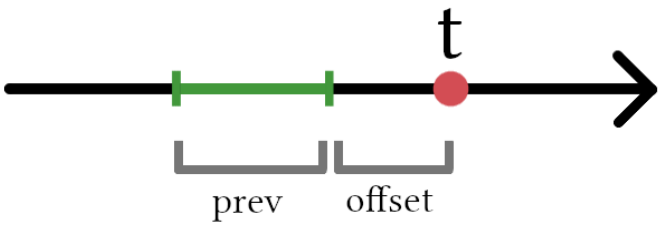
Un'altra forma di preprocessing che si è pensata di effettuare è l'applicazione di un logaritmo alla variabile portata, poiché molto "sparsa" (cioè molto costante su valori bassi). Questa operazione non è poi stata confermata poiché le previsioni generano spesso valori molto bassi, non andando a prevedere ciò che conta, cioè i forti picchi.

Inoltre, nonostante la correlazione notata tra le 5 feature di pioggia, è stato deciso di mantenerle tutte, in quanto, anche se sporadicamente, possono aiutare nel migliorare le previsioni, specialmente in modelli complessi come le reti neurali.

Vincoli

Il problema di base impone di prevedere il livello d'acqua e la portata di uscita con una settimana di anticipo. Chiameremo questo vincolo temporale **offset**. A questo problema si aggiunge lo studio sulla sua variazione; assumerà i seguenti valori: 1, 3, 5, 7, 14, 21, 28.

Una questione importante è data dal fatto che, per allenare un modello di regressione di machine / deep learning, è necessario che i dati in input abbiano tutti la stessa dimensione. Quindi, bisogna determinare quanti giorni, precedenti al giorno **t** che si vuole predire, si vuole utilizzare. Chiameremo questo vincolo **prev**. Verranno testate diverse combinazioni di prev, per capire quale è il migliore per ciascun offset. Assumerà i seguenti valori: 1, 3, 5, 7, 14, 21, 28, 42, 56, 70, 84, 98, 112. La scelta di questi numeri è dettata dalla loro distribuzione, simile agli offset, con in aggiunta più valori a intervalli di due settimane.

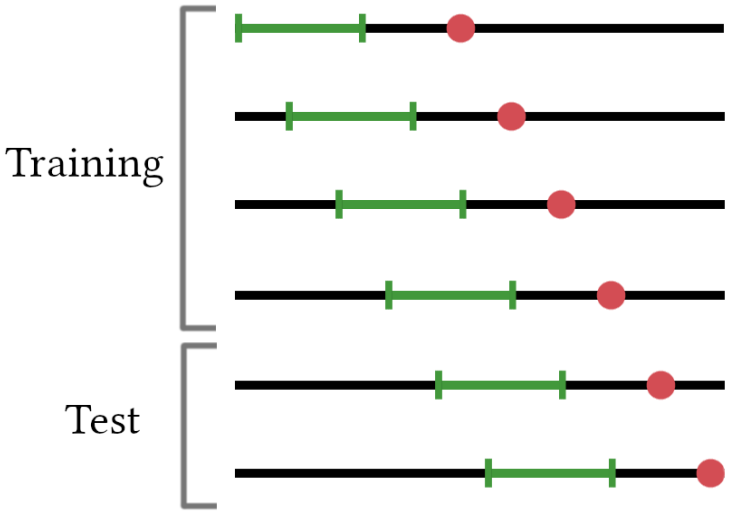


Schema riassuntivo della struttura del problema.

Costruzione del dataset per i modelli

Il dataset è stato costruito facendo scorrere una finestra sulle serie storiche: ogni osservazione verrà usata come target (y), usando come dati in input (X) i giorni precedenti (a seconda di quanto valgono prev e offset).

Per quanto riguarda la suddivisione in training e test set, è stata scelta come dimensione di training l'80% dei dati e di test il 20%. Il test set, a sua volta, verrà diviso in modo equo tra validation e test.



Schema riassuntivo sulla suddivisione del dataset.

È da tenere a mente che, dovendo testare tutte le combinazioni di offset e prev, sono stati salvati dei file di training, validation e test per ciascuna di esse.

Inoltre, come si può intuire dall'immagine, a seconda della dimensione di offset e prev, alcuni dati vengono persi. Infatti, le prime osservazioni non possono essere usate come target, non avendo dati precedenti a sufficienza (varia a seconda di prev e offset).

4. Metodologia

La predizione nel futuro del valore di livello dell'acqua, o di portata, viene visto come un problema di **regressione**. Sono stati quindi selezionati diversi metodi, a complessità crescente ma con caratteristiche diverse, che svolgessero questo tipo di analisi.

Il procedimento generale, per ogni tipo di modello, è stato quello di individuare inizialmente la struttura e gli iperparametri ottimali; successivamente, si è andati a determinare, per ogni offset, qual è il miglior valore prev da utilizzare, in modo da avere predizioni più precise. Alla fine di questo processo, si avranno, per ogni tipo di modello, 14 diversi modelli (7 offset per 2 target). I risultati sul validation set verranno successivamente comparati per determinare qual è il migliore.

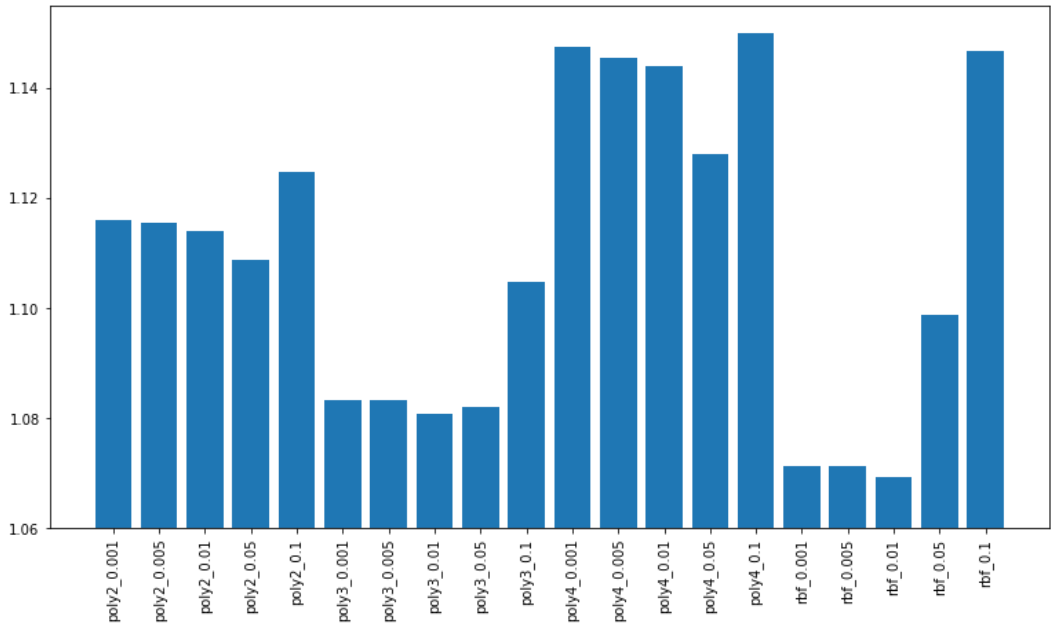
La metrica scelta per valutare le performance è il **MAE** (mean absolute error), in quanto **direttamente interpretabile**: infatti, è possibile ottenere il numero di metri di cui si sbaglia in media, nel caso del target livello d'acqua (o di m^3/s nel caso della portata di uscita).

SVR (Support Vector Regression)

Il primo metodo utilizzato è una regressione tramite SVR di scikit learn [1]. La Support Vector Regression è un tipo di Support Vector Machine che supporta regressione di tipo lineare e non. Lo scopo è di adattare quante più istanze possibili all'interno delle soglie, limitando le violazioni dei margini che sono rappresentate dal parametro epsilon. Quindi, a differenza della regressione lineare, che cerca di minimizzare l'errore tra la predizione e l'osservazione, SVR cerca di non fare eccedere l'errore oltre le threshold.

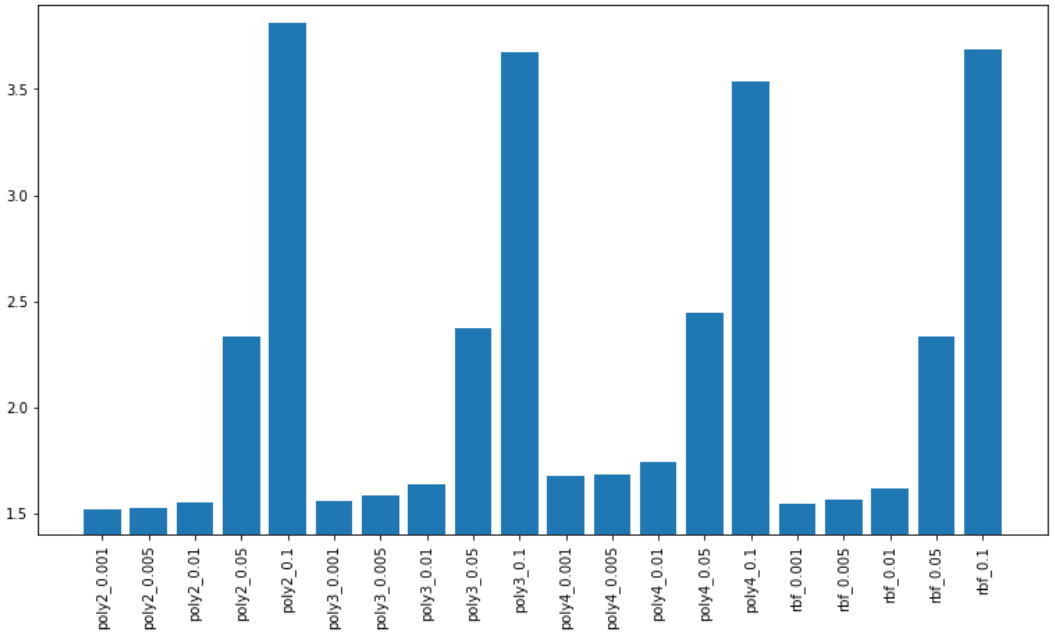
Al fine di trovare i migliori iperparametri per allenare il modello, è stata applicata una grid-search con diversi valori di kernel, grado ed epsilon. Le diverse combinazioni sono state applicate ad entrambe le feature target, su alcuni prev e offset prefissati (mediando i MAE risultanti).

Kernel	Degrees	Epsilon
Poly	2, 3, 4	0.001, 0.005, 0.01, 0.05, 0.1
Rbf	-	0.001, 0.005, 0.01, 0.05, 0.1



*Risultati per target **livello**.*

Per quanto riguarda il livello, si nota che con un epsilon elevato i risultati sono scarsi; inoltre, il kernel rbf ottiene risultati leggermente migliori nella maggior parte dei casi. A valle di questa analisi, si opta per l'architettura con **kernel rbf** ed **epsilon 0.01**, non solo per i risultati ma anche per il valore non troppo eccessivo del parametro epsilon.

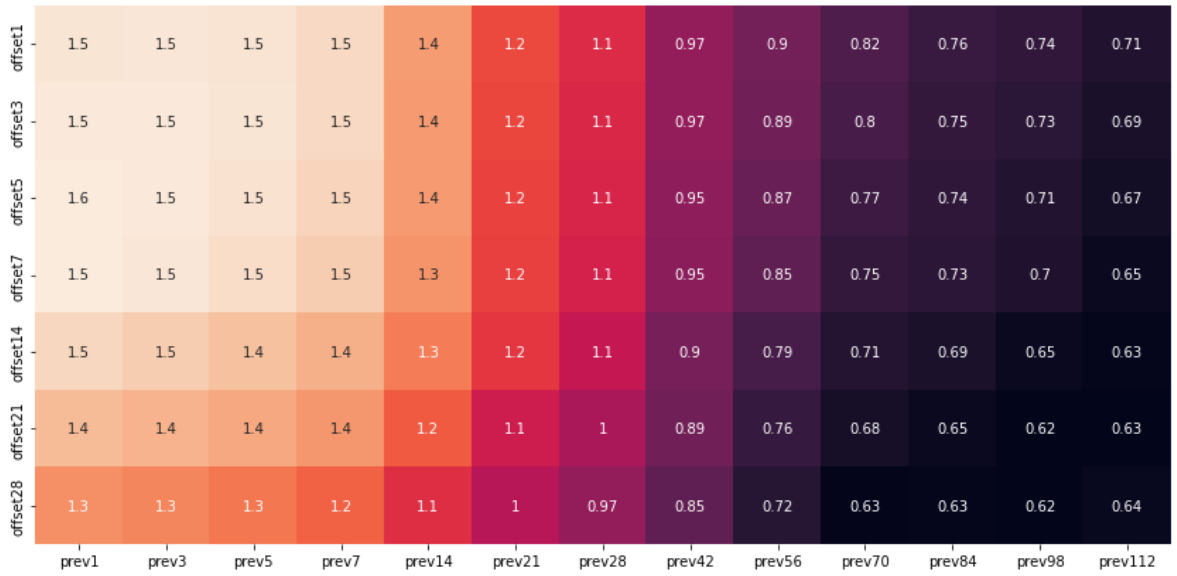


*Risultati per target **portata**.*

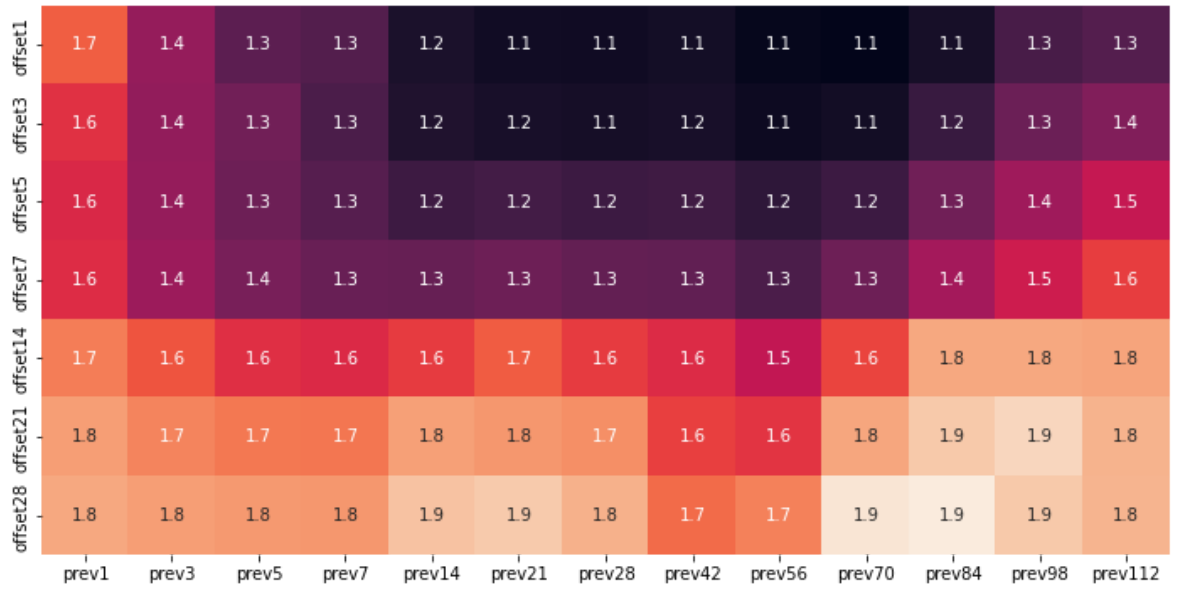
Per la portata, si nota come il fattore più influente sia il parametro epsilon. Infatti, al diminuire di epsilon si ottengono risultati migliori. Ad occhio, è facile notare come all'aumentare del parametro degree dei modelli con kernel polinomiale diminuiscano le performance. In generale, il miglior modello ha **epsilon 0.001** e **kernel polinomiale di 2° grado**.

A questo punto, al fine di individuare, per ogni offset, il prev in grado di generare il modello più performante, vengono costruite e testate nuove istanze per ogni combinazione.

Per quanto riguarda il livello, il MAE tende a decrescere continuamente all'aumentare del prev, raggiungendo il valore più basso assoluto a 98. Per la portata, invece, si ottengono risultati migliori con prev tra 14 e 84 con offset piccoli, di 1 o 3.



*MAE per ogni combinazione di prev e offset, sulla variabile di target **livello**.*

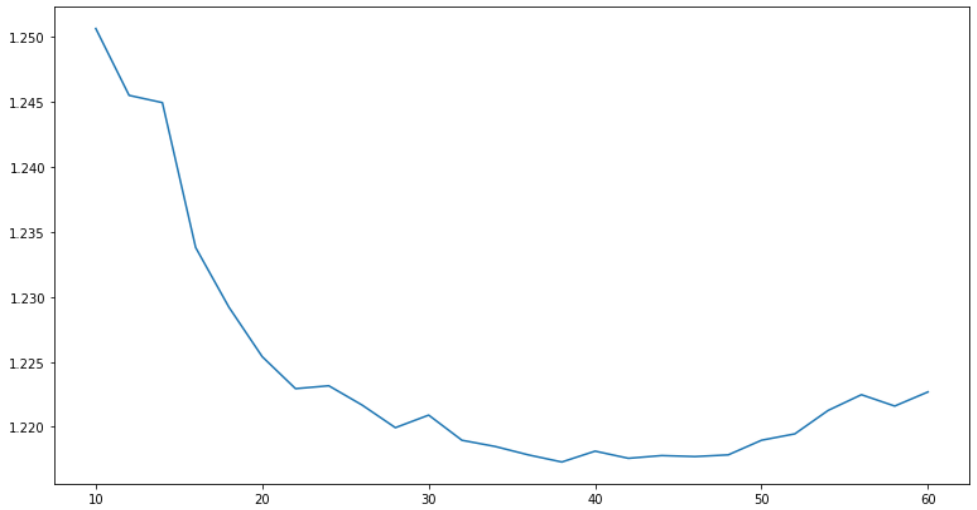


*MAE per ogni combinazione di prev e offset, sulla variabile di target **portata**.*

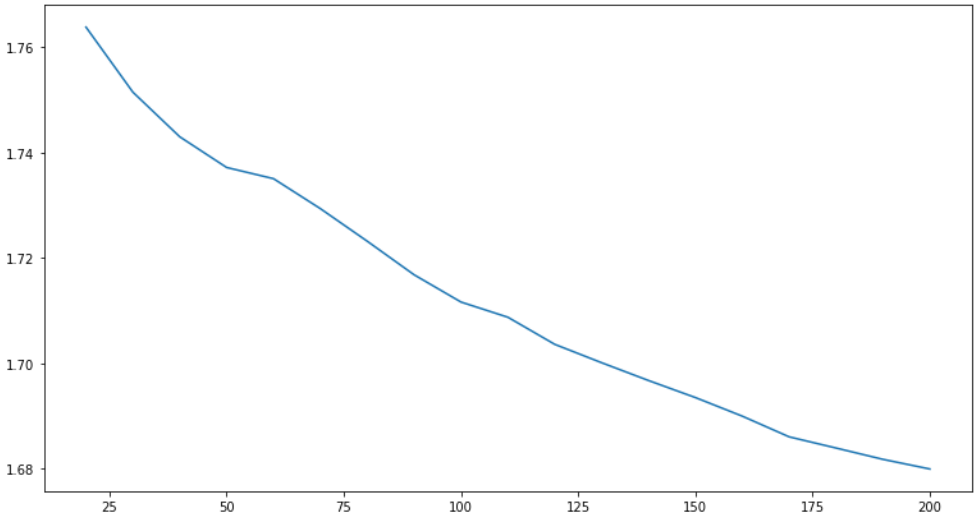
KNN (K-Nearest Neighbors)

Questo metodo si basa sull'algoritmo di classificazione dei K-Nearest Neighbors [2], che può essere utilizzato anche per problemi di regressione. Nella regressione KNN, viene calcolata una media pesata dei valori dei k vicini, pesata per l'inverso della loro distanza.

Per determinare l'iperparametro K (numero di vicini), viene effettuata una grid-search per i due diversi target, su alcuni prefissati offset e prev.



*MAE medio per i K tra 10 e 60 per il **livello**.*

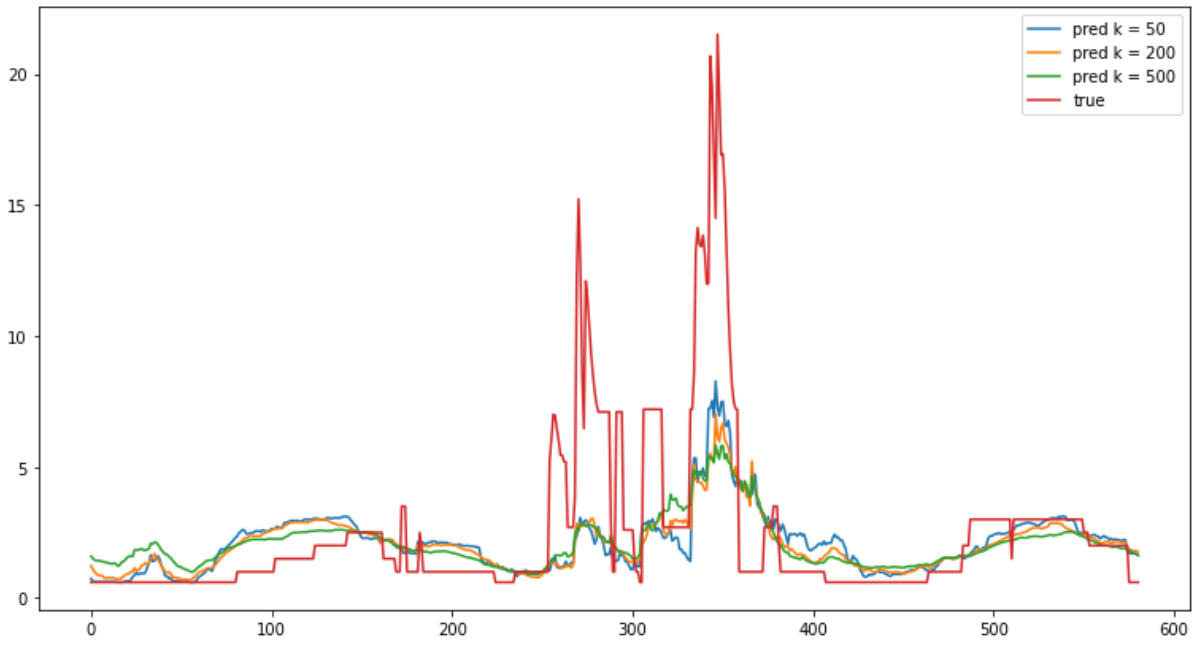


*MAE medio per i K tra 20 e 200 per la **portata**.*

Mentre per il livello il miglior K è pari a 38, lo stesso non si può dire per la portata. Infatti, si presenta un continuo miglioramento del MAE.

Per comprendere se questo fenomeno fosse effettivamente corretto, è stata effettuata una previsione per osservarne la curva.

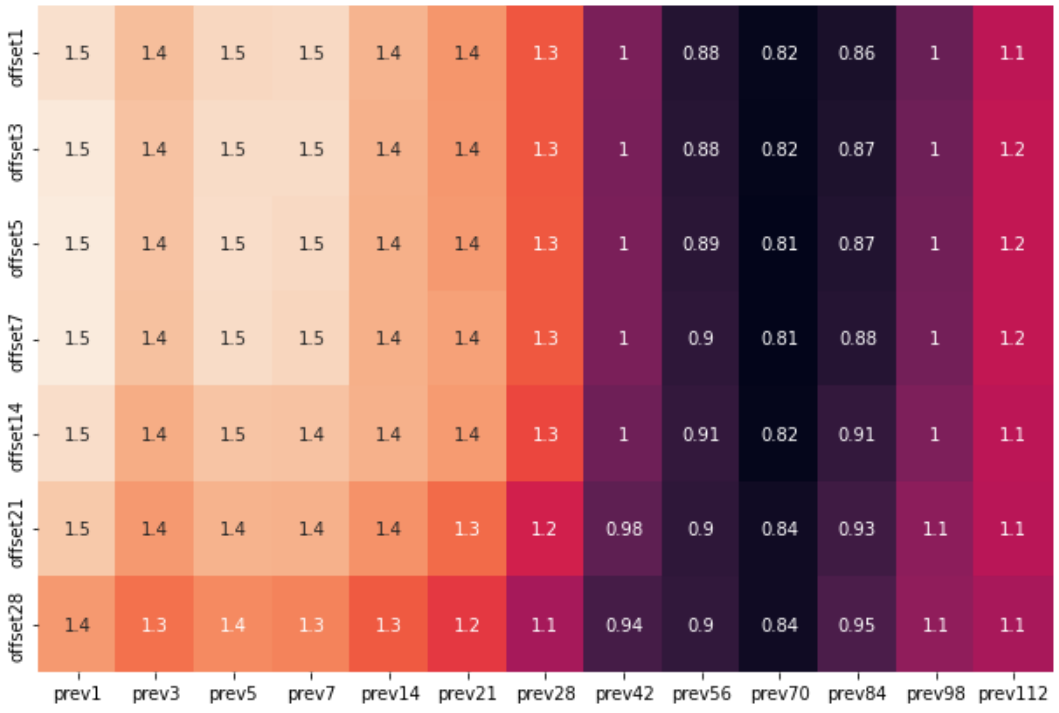
Ciò che si nota osservando le previsioni è che, a causa della sparsità della feature target portata (problema che abbiamo cercato di indirizzare), all'aumentare del numero K di neighbors la previsione tende ad appiattirsi, portando ad un miglioramento del MAE ma ad una drastica diminuzione della qualità e della significatività della predizione. Infatti, il valore aggiunto ci viene dato quando riusciamo a prevedere uno dei rari picchi di valore nella serie e non consegnando ovvietà.



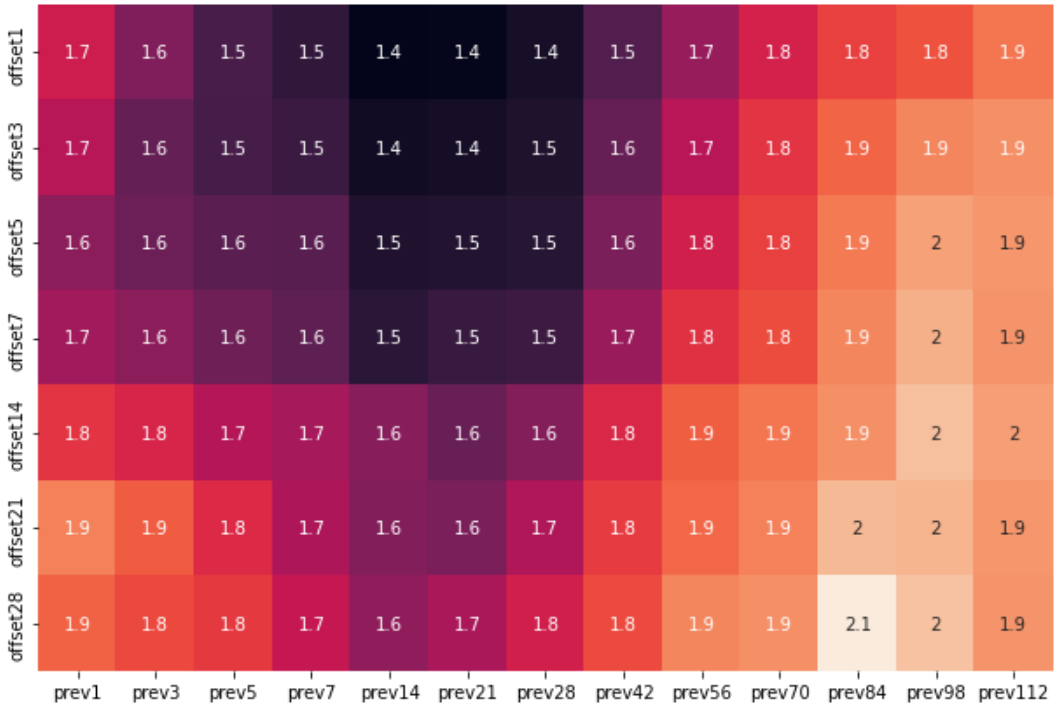
Predizione rispetto al valore reale (rosso) sul validation set, calcolata con k=500 (verde), k=200 (arancione) e k = 50 (blu), offset 1 e prev 21

Dunque, data la difficoltà riscontrata nel determinare il miglior numero K di neighbors, si è scelto un valore arbitrario pari a 50.

Una volta scelto il K, si è proseguito col determinare i migliori prev:



MAE per ogni combinazione di prev e offset, sulla variabile di target ***livello.***



MAE per ogni combinazione di prev e offset, sulla variabile di target ***portata***.

Per il livello, i risultati non sembrano essere influenzati dall'offset. In tutti i casi, i migliori risultati si hanno con prev 70. Per la portata, i migliori risultati sono ottenuti con un numero di prev compreso tra 14 e 28.

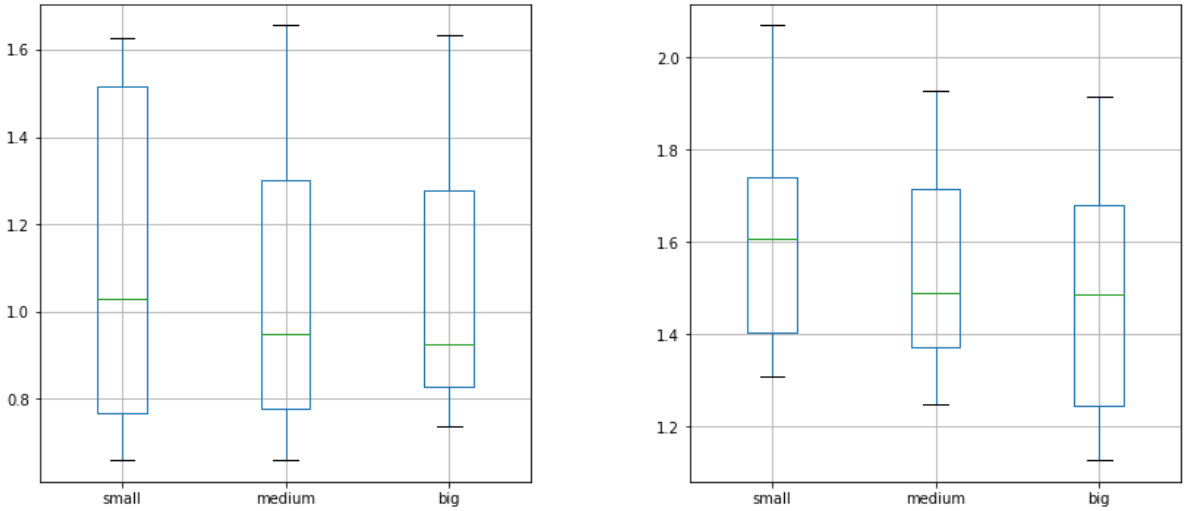
Reti Neurali

Gli altri metodi si basano su reti neurali, sviluppati tramite Keras [3]. La metrica utilizzata come funzione di loss durante l'addestramento è l'MSE (mean squared error), in modo da penalizzare maggiormente gli errori più elevati.

Feedforward Neural networks

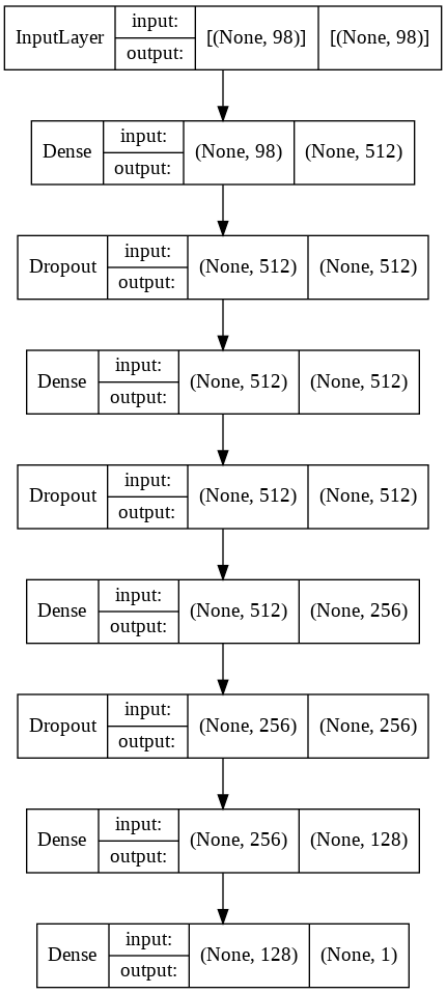
Questo tipo di reti neurali è piuttosto semplice, essendo composte da layer densi e opzionalmente di regolarizzazione. Per determinare l'ampiezza e la profondità, sono stati effettuati dei test con tre diverse architetture, a complessità crescente, prefissando alcuni prev (1, 7, 28, 70, 112) e alcuni offset (1, 7, 28).

Per entrambe le variabili target, l'architettura migliore risulta essere la più complessa. È da specificare che aumentare ulteriormente la complessità non ha portato ad alcun miglioramento.



Distribuzione del MAE per i diversi modelli, per livello (sx) e portata (dx).

Di seguito, è la rete risultante composta da layer densi con funzione di attivazione ReLU, dropout e output function lineare.

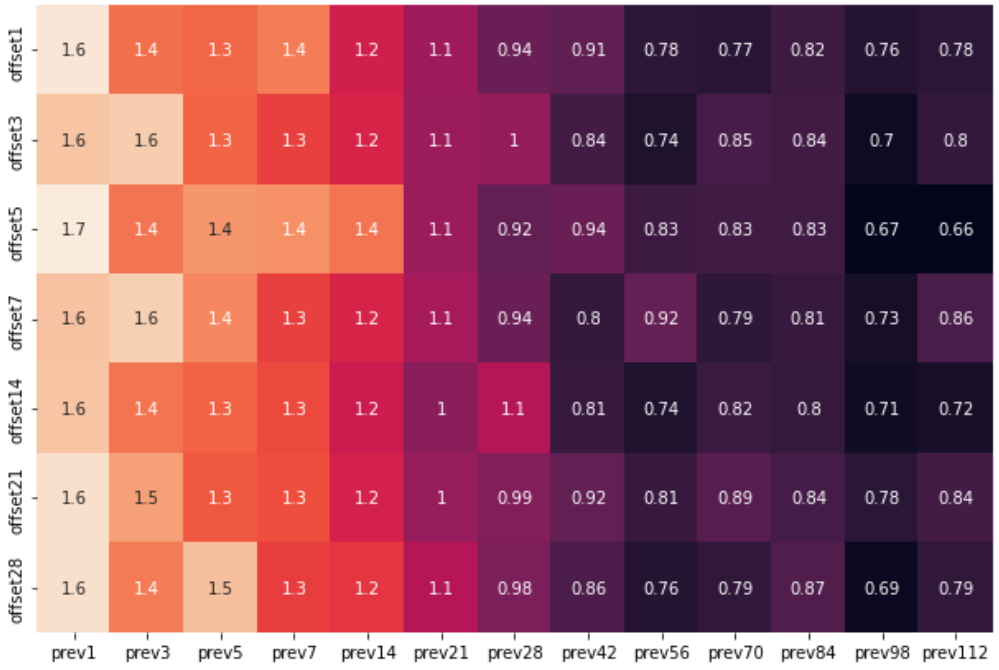


Il training della rete, per entrambe le variabili target, è stato effettuato con i seguenti iperparametri:

- batch size: 16
- loss: mean squared error
- epoche: 100
- early stopping, con pazienza: 3

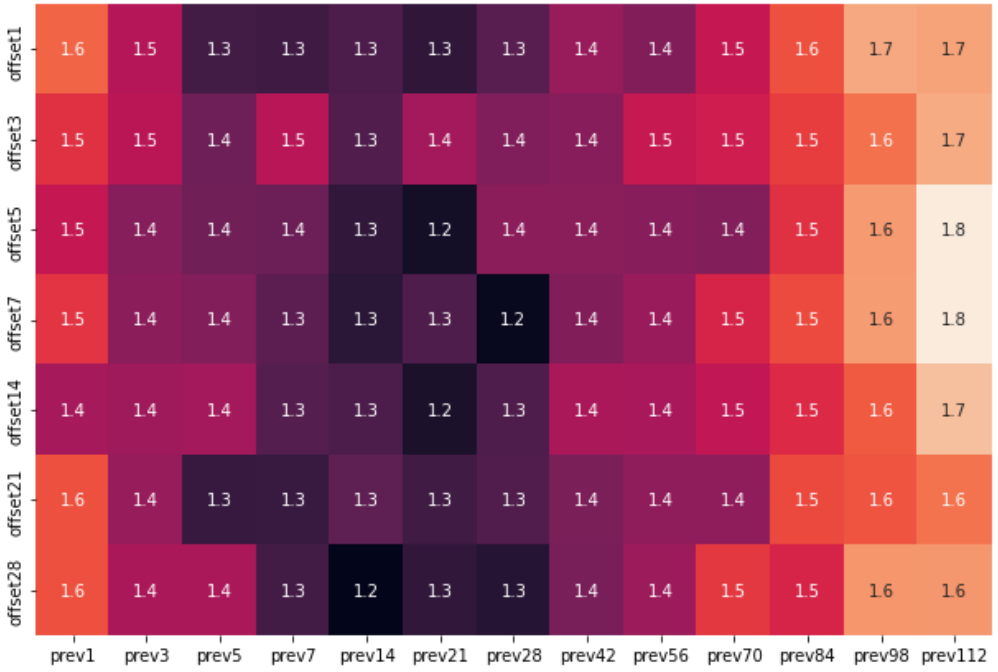
Come per i precedenti metodi, viene testata ogni combinazione di prev e offset. Inoltre, ogni combinazione viene eseguita tre volte (e ne viene fatta una media del MAE), in modo da ovviare alla stocasticità* della rete neurale.

* *Nonostante siano stati fissati dei seed, esiste comunque una componente derivante dall'uso della GPU.*



MAE per ogni combinazione di prev e offset, sulla variabile di target ***livello****.*

Per quanto riguarda il livello, il MAE tende a decrescere all'aumentare del prev, raggiungendo il valore più basso generalmente a 98.



MAE per ogni combinazione di prev e offset, sulla variabile di target ***portata***.

Per la portata, invece, si ottengono risultati migliori con prev tra 14 e 28.

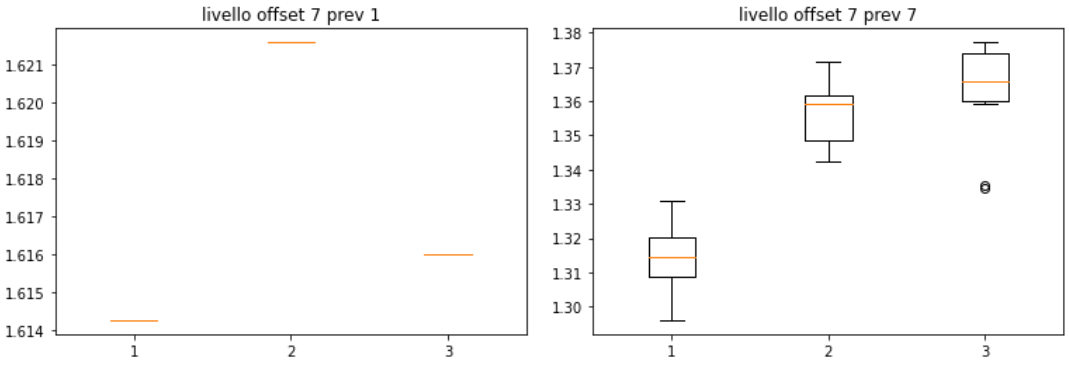
Recurrent Neural Networks (LSTM)

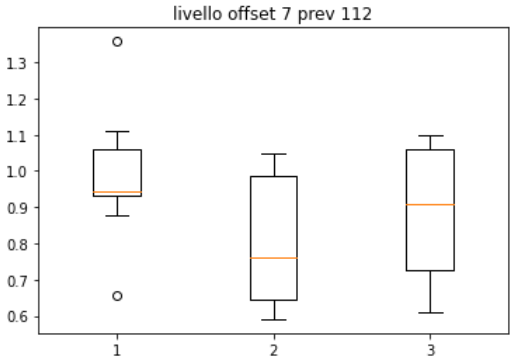
A differenza delle reti neurali feedforward, le LSTM posseggono connessioni di feedback in grado di apprendere pattern temporali nelle serie, quindi il suo utilizzo si presta particolarmente a questo use-case.

Al fine di scegliere la migliore architettura possibile, sono stati effettuati dei test con modelli a diversi livelli di profondità:

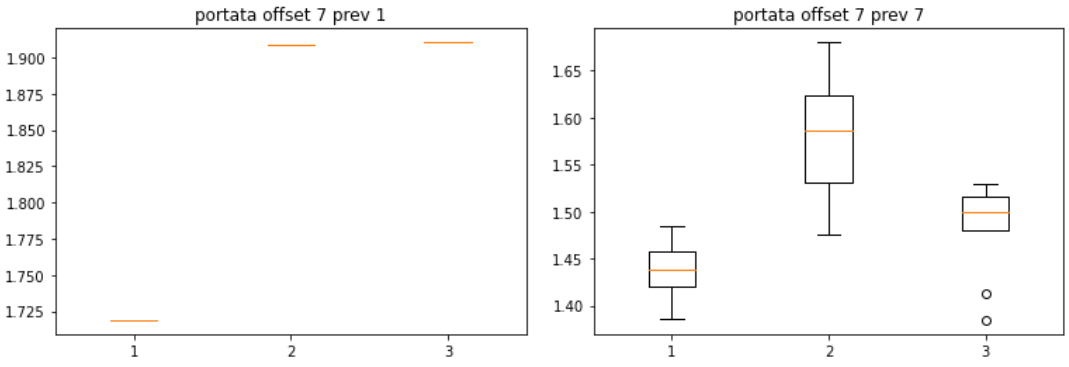
- 1 layer 128 neuroni
- 2 layer 128, 64 neuroni
- 3 layer 128, 64, 32 neuroni

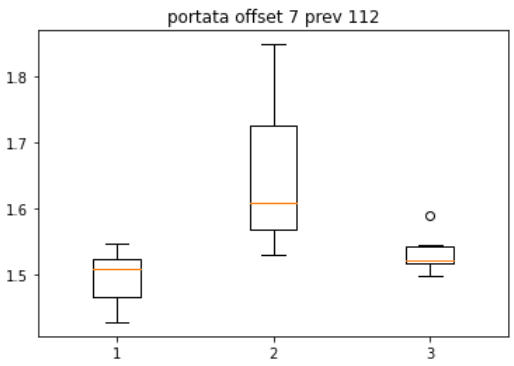
Fissando l'offset, per ogni modello sono state effettuate varie predizioni con 3 diversi prev: 1, 7 e 112. Lo scopo è valutare la **relazione** tra le performance e la grandezza dei prev, al fine di poter meglio decidere quale **architettura** utilizzare durante la successiva fase di addestramento per i due diversi tipi di target feature.





Le performance tra i 3 modelli con prev 1 sono abbastanza simili (notare asse y); con prev 7 inizia a notarsi una differenza tra il modello a 1 layer e gli altri più profondi; infine, con prev 112, nonostante una deviazione standard piuttosto elevata, il modello con 2 layer ottiene una performance media migliore. Ipotizzando che all'aumentare dei dati in input al modello, e quindi del numero di prev, le performance tendono a crescere, viene scelta come architettura per la target feature **livello** il modello a 2 layer.

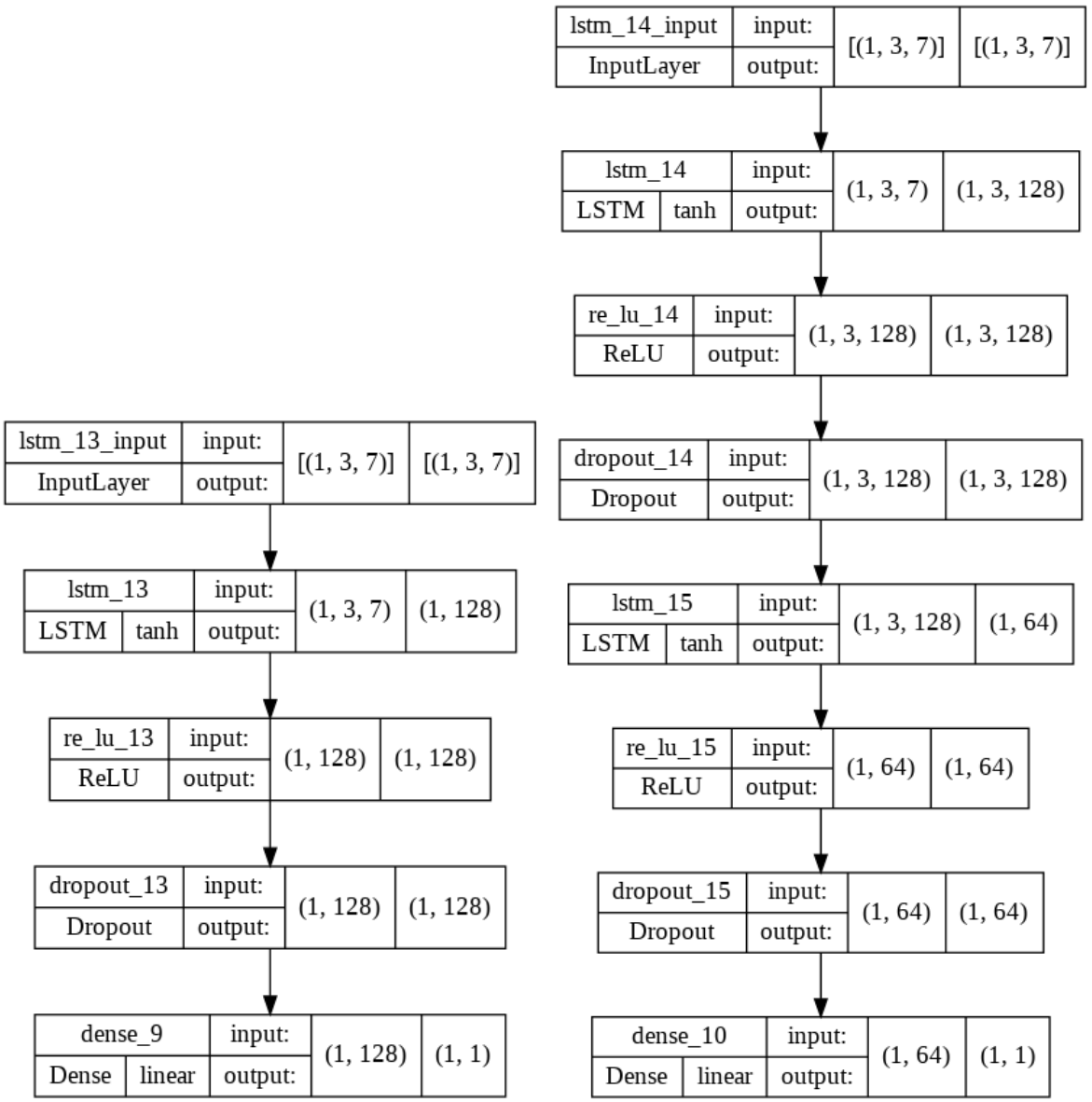




Nel caso della feature target **portata** si nota facilmente come il modello con 1 solo layer ottenga un MAE più basso in tutti e 3 i test con prev a 1, 7 e 112.

Una possibile giustificazione di questo comportamento può essere dovuta alla logica delle LSTM, ovvero reti composte da più layer riescono a mantenere una memoria più lunga e di conseguenza performare meglio con più dati.

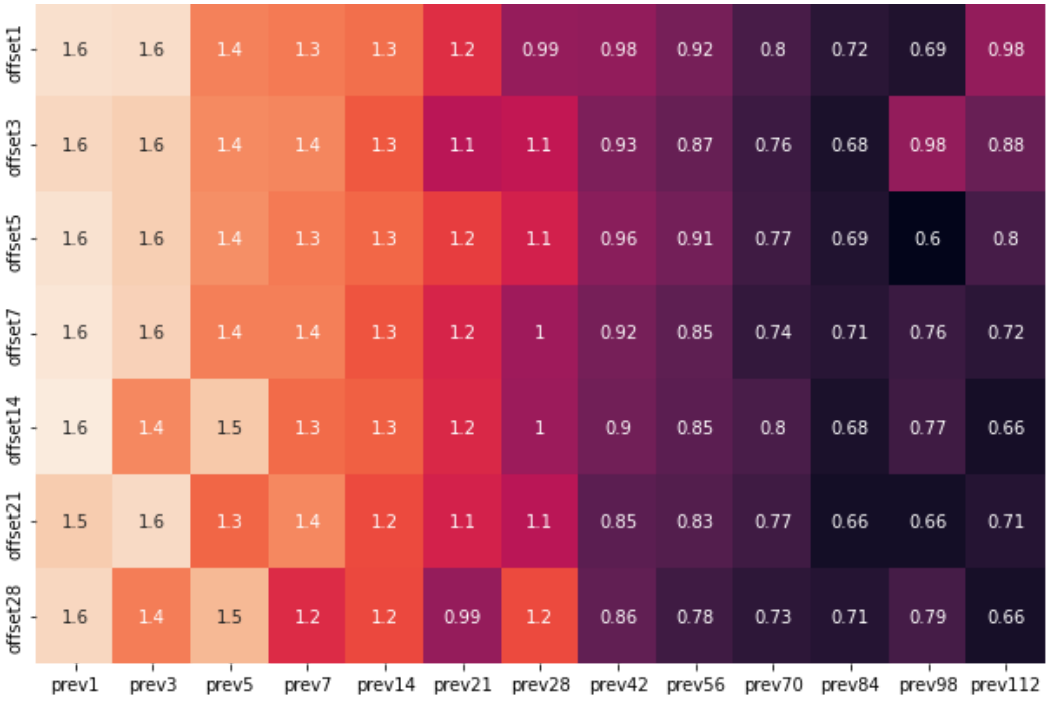
Sono quindi scelte due diverse architetture per le due target feature: una rete a 2 layer per il target **livello** ed una rete con un solo layer per il target **portata**.



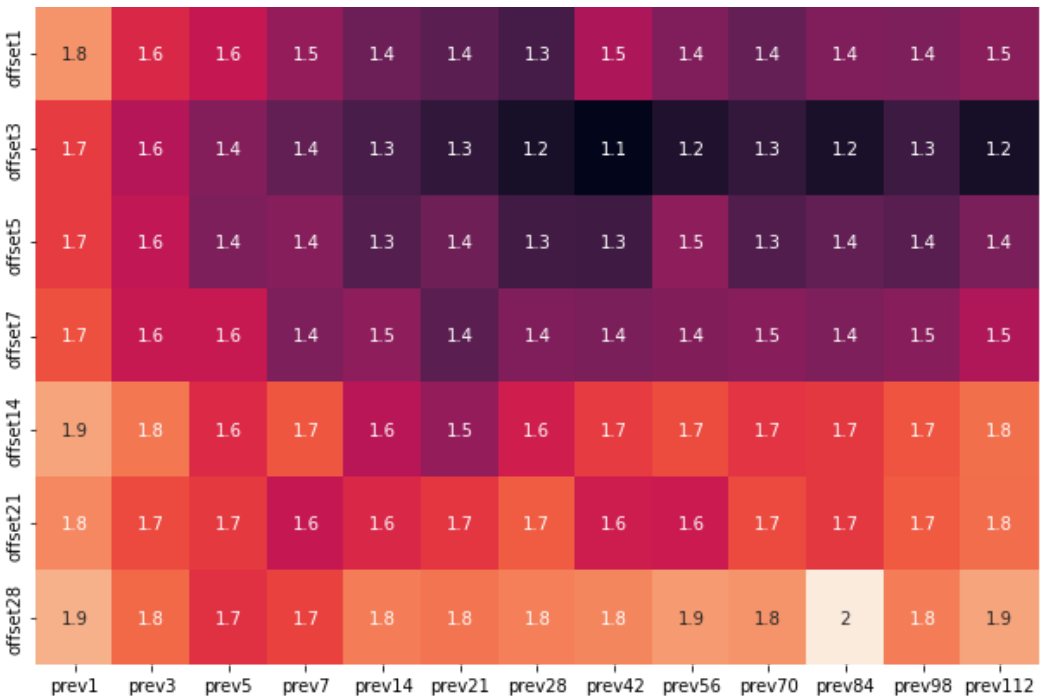
Preview dei modelli, rispettivamente con 1 layer e con 2 layers. La shape dell'input è composta da [1, prev, numero features]

Come in precedenza, l'addestramento del modello viene effettuato per ogni combinazione possibile, al fine di individuare il miglior prev per ogni offset. Eseguendo ogni combinazione tre volte, quindi computando una media del MAE, in modo da ovviare alla stocasticità* della rete neurale.

** Nonostante siano stati fissati dei seed, esiste comunque una componente derivante dall'uso della GPU.*



*MAE per ogni combinazione di prev e offset, sulla variabile di target **livello**.*



*MAE per ogni combinazione di prev e offset, sulla variabile di target **portata**.*

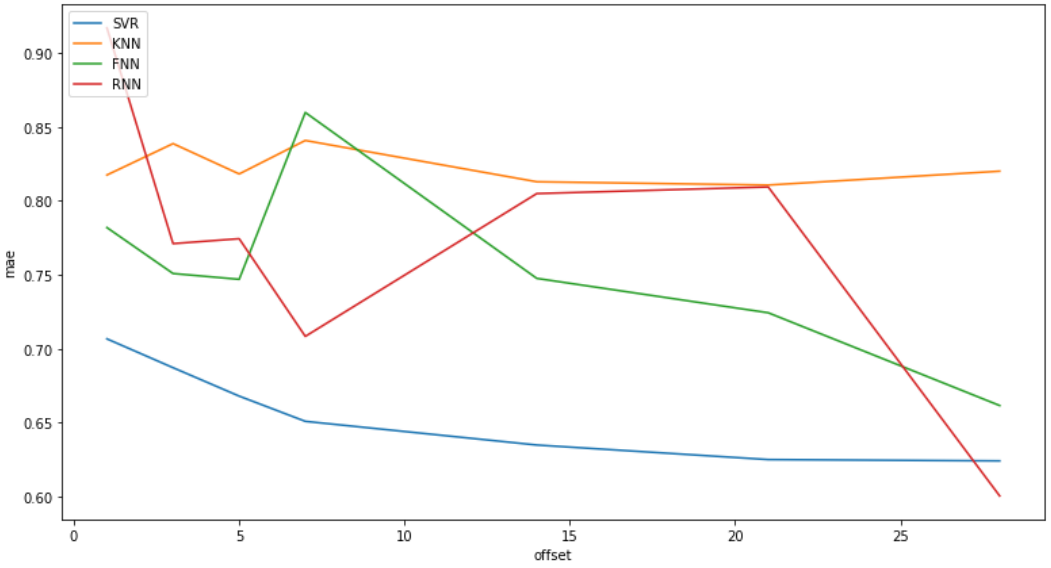
Per quanto riguarda il **livello**, il MAE tende a decrescere all'aumentare del prev, raggiungendo il valore più basso generalmente a 98. Per la **portata**, il MAE tende a decrescere all'aumentare del prev ma con valori di offset bassi, raggiungendo il valore più basso con offset a 3.

5. Risultati

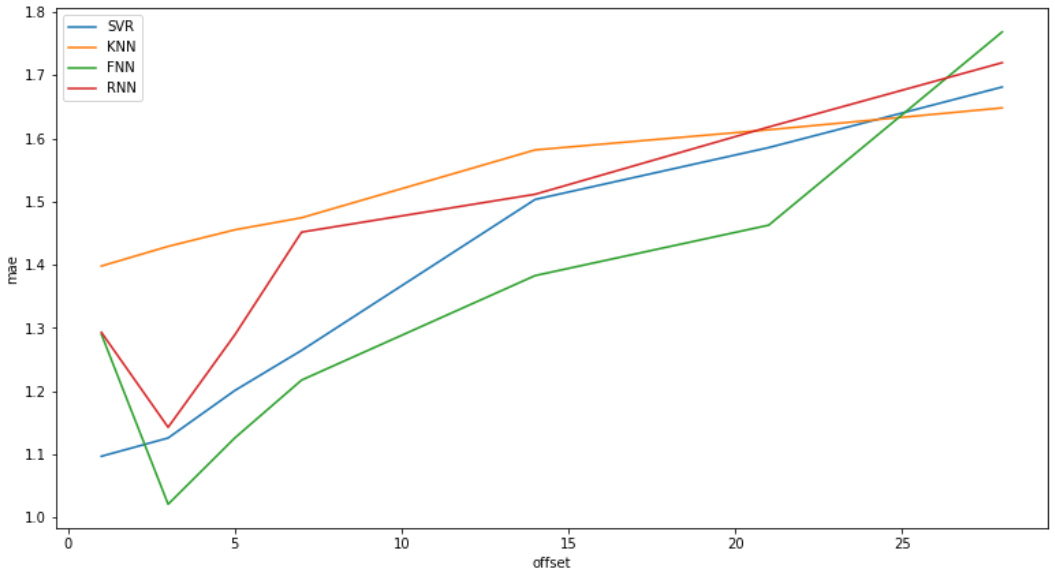
Confronto metodi

I 4 metodi vengono comparati tramite MAE sul validation set.

Nel target **livello** i risultati migliori sono chiaramente ottenuti dalla **SVR**, che, nonostante la sua semplicità, riesce ad ottenere un MAE discreto, ad esclusione dell'offset 28. Nel target **portata**, invece, è la **feedforward neural network** ad ottenere MAE inferiori, ad esclusione degli offset 1 e 28.

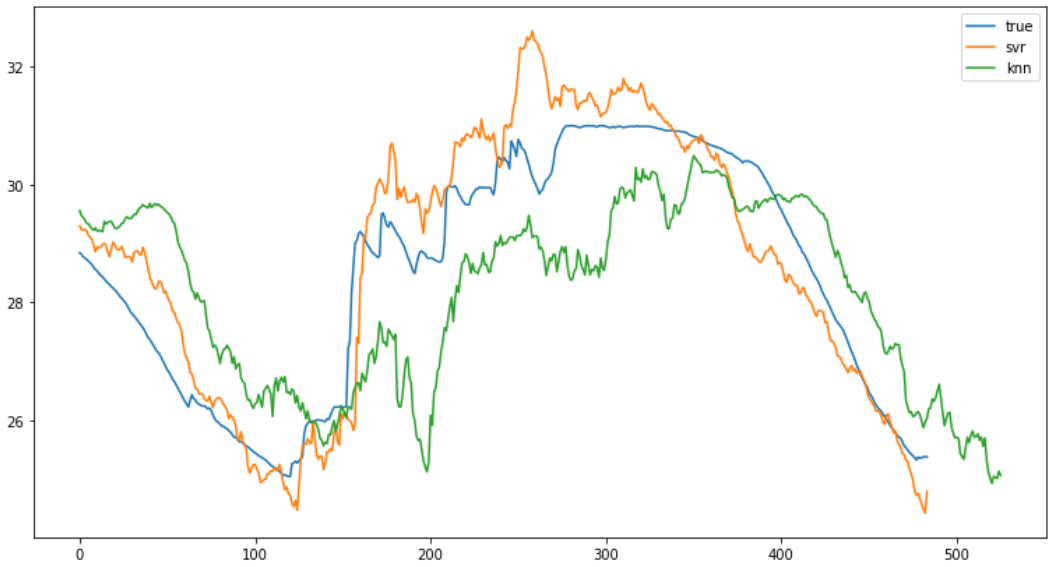


*MAE di ciascun modello su tutti gli offset, sul validation set, per il target **livello**.*



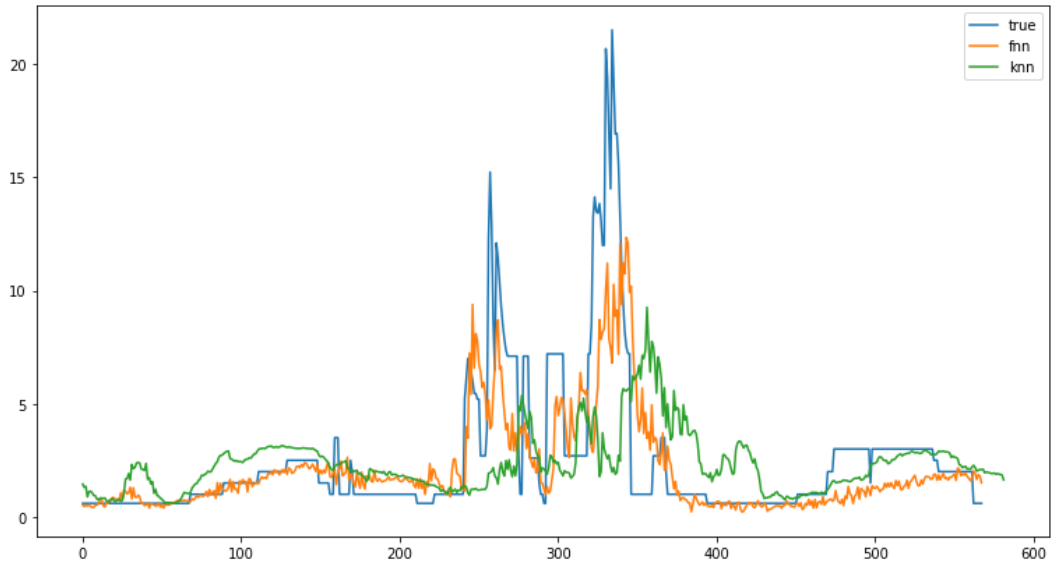
MAE di ciascun modello su tutti gli offset, sul validation set, per il target ***portata***.

Al fine di mostrare in cosa effettivamente consista avere un MAE inferiore, vengono mostrati due esempi. Il primo riguarda le predizioni sul livello, relative al migliore e peggiore modello, ovvero **SVR** e KNN. Si nota che la curva di predizione del KNN si allontana maggiormente dalla curva di valori reali.



Predizione sul livello, con un offset arbitrario di 7,
di ***SVR (MAE = 0.651)*** *e KNN (MAE = 0.811)*

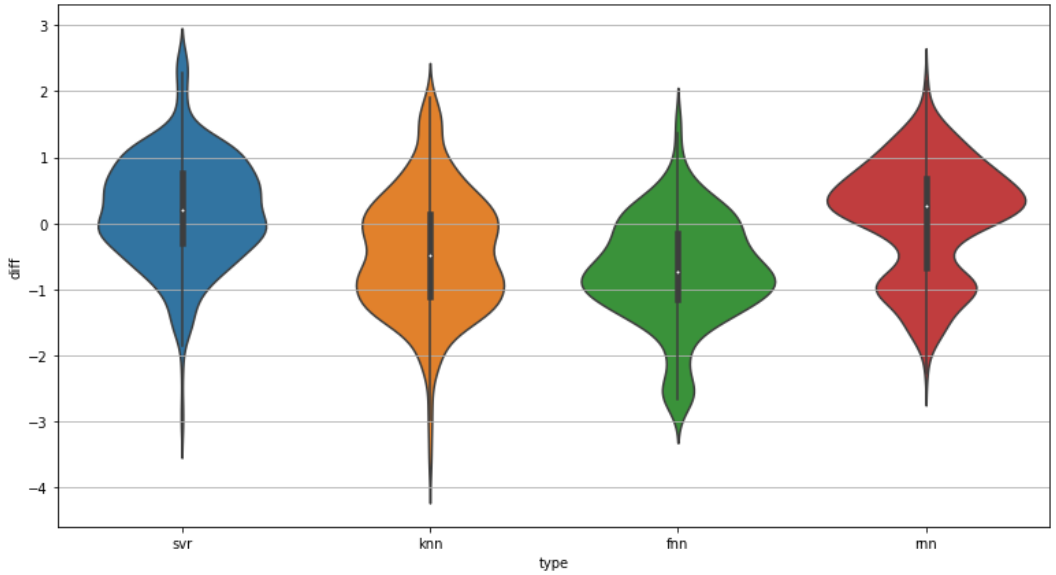
Altro esempio è quello sulla portata, con predizioni relative ai modelli **FNN** e KNN, rispettivamente il migliore ed il peggiore. Si nota come la predizione della feedforward rispetti meglio i due picchi caratteristici della serie rimanendo invece piatta il resto del tempo.



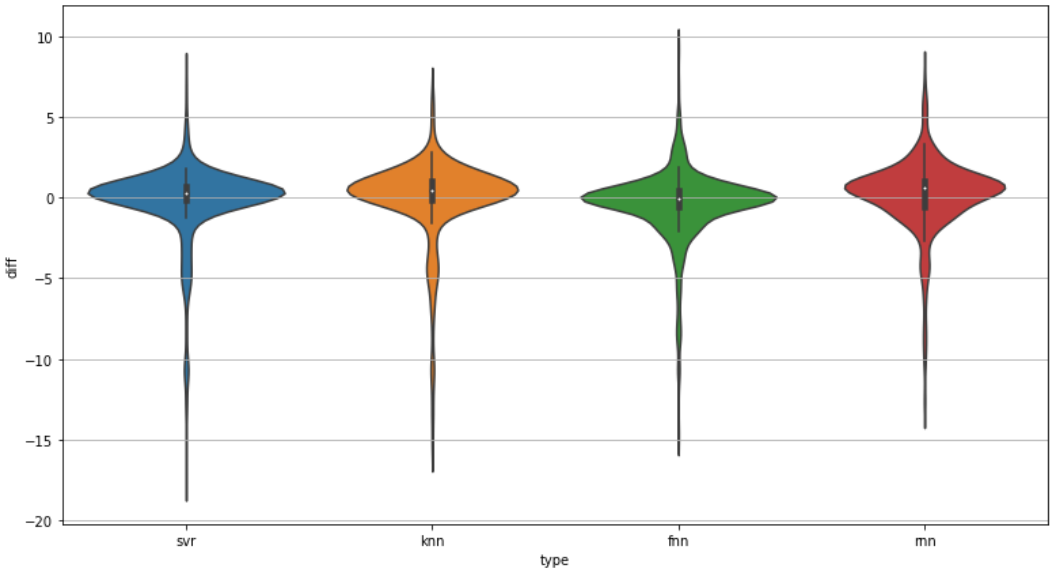
Predizione sulla portata, con un offset arbitrario di 7,
*di **FNN (MAE = 1.218)** e KNN (MAE = 1.475)*

Una questione importante nella gestione di un bacino idrico è la differenza tra una sottostima e una sovrastima, che può giocare un ruolo fondamentale nel caso di siccità / surplus di acqua.

Per osservare questo fenomeno è possibile mostrare le distribuzioni degli scarti dei 4 modelli, fissando arbitrariamente l'offset a 7 (ovviamente rispecchiano i MAE, essendo quest'ultimo semplicemente la media del valore assoluto di questi errori).



*Scarti di ciascun modello con offset 7, sul validation set, per il target **livello**.*



*Scarti di ciascun modello con offset 7, sul validation set, per il target **portata**.*

Osservando i grafici degli scarti, si può notare che SVR e RNN generalmente sovrastimano il valore di **livello**, quando commettono un errore; al contrario, KNN e FNN sottostimano. Questa informazione può essere utile nella gestione del bacino idrico, ma non avendo informazioni di dominio aggiuntive, verrà selezionato semplicemente il modello con il MAE più basso.

Per la **portata** si osserva che gli scarti della FNN (modello migliore) hanno media quasi pari a 0, ma alle volte il modello commette degli errori più grossi di quanto la RNN farebbe. Anche in questo caso, il committente potrebbe preferire un modello che sbaglia entro un certo range, ma non avendo informazioni di dominio verrà di nuovo selezionato il modello a MAE minore.

In definitiva, in seguito a tutte queste analisi, i modelli selezionati sono:

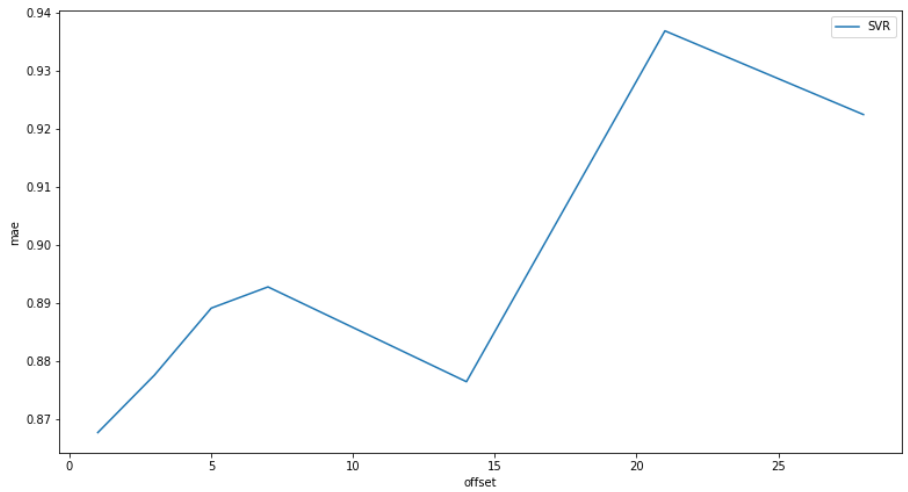
- per il target **livello**, una **Support Vector Regression** con kernel rbf ed epsilon 0.01
- per il target **portata**, una **Feed-Forward Neural Network** a 4 layer densi.

Risultati sul test set

Una volta selezionati i migliori modelli sulla base delle performance sul validation set, vengono effettuate delle previsioni sul test set, per determinare la loro accuratezza finale.

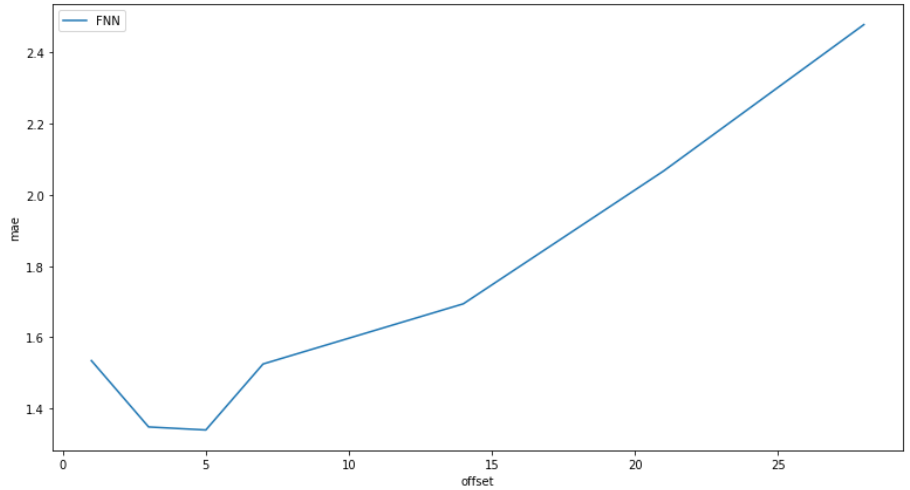
Come ci si poteva aspettare, la qualità delle previsioni sul test set cala all'aumentare del valore di offset.

Nel caso della variabile target livello, si notano due piccole correzioni che compromettono la monotonia all'aumentare degli offset. Queste inversioni possono essere facilmente giustificate da una variazione molto piccola del MAE, alla seconda cifra decimale, che è verosimilmente rumore.



offset	mae
1	0.868
3	0.878
5	0.889
7	0.893
14	0.876
21	0.937
28	0.923

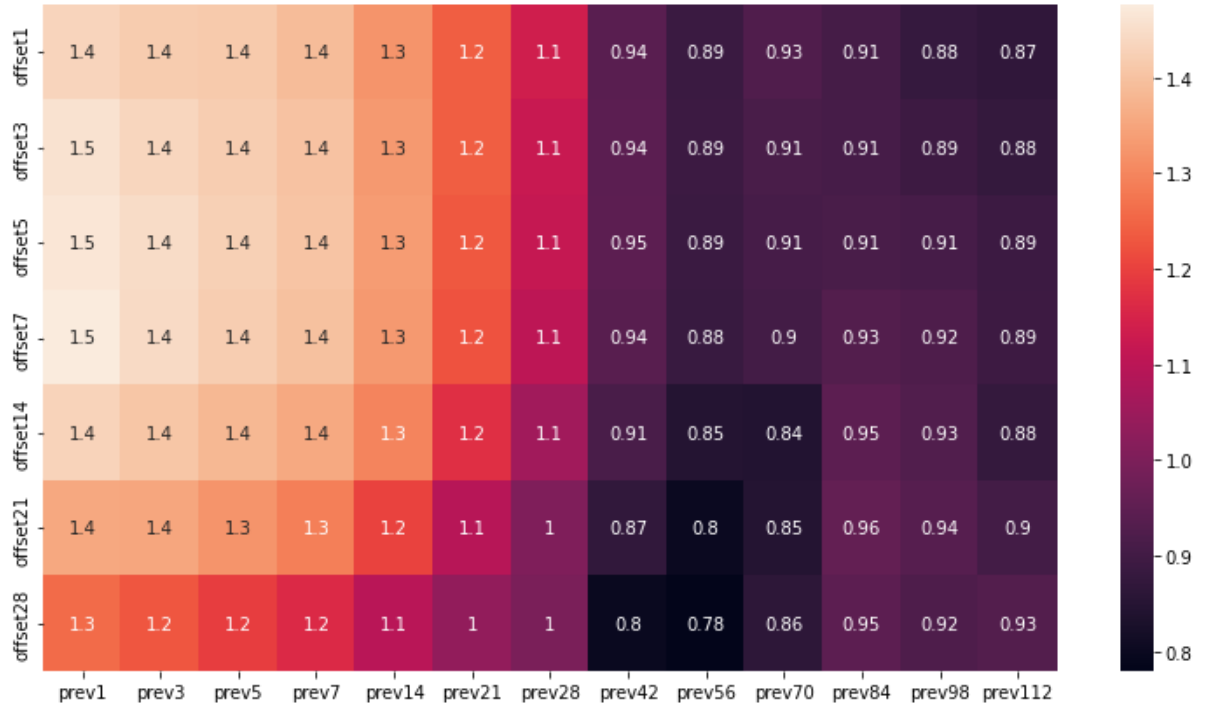
*Risultati sul target **livello**, con il modello SVR.*



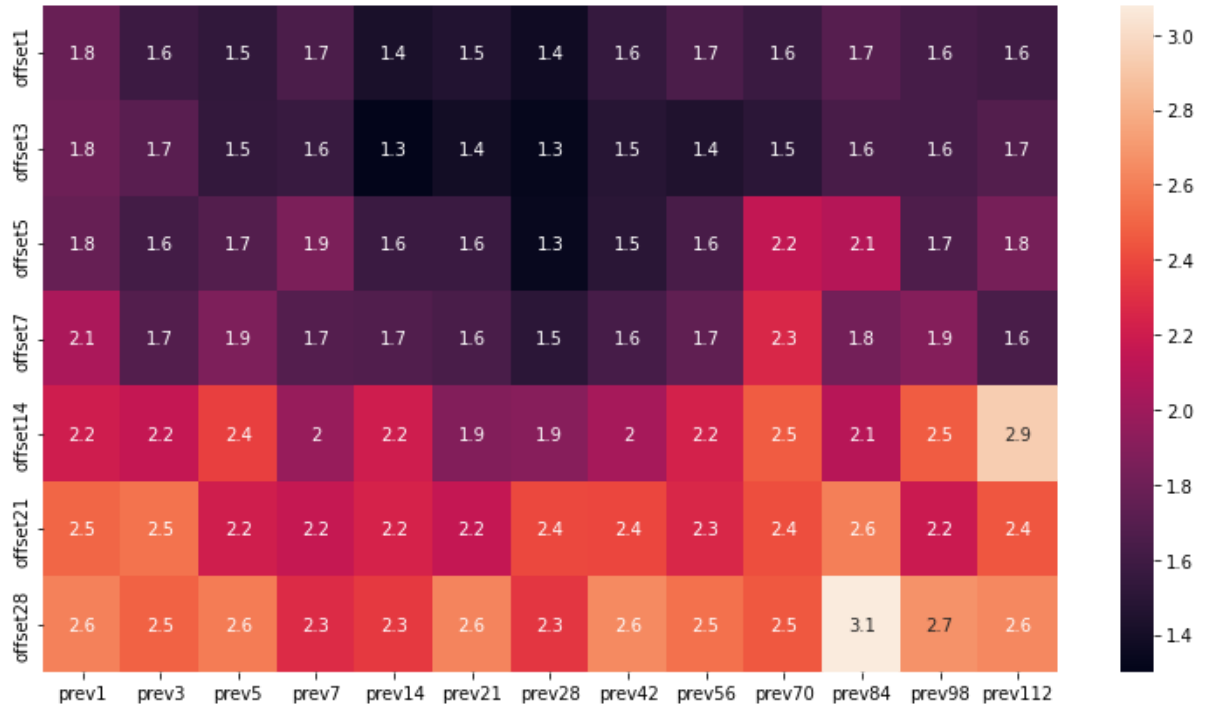
offset	mae
1	1.535
3	1.349
5	1.340
7	1.525
14	1.694
21	2.067
28	2.477

*Risultati sul target **portata**, con il modello FNN.*

Per completezza, vengono presentati anche i risultati con tutti i restanti prev.



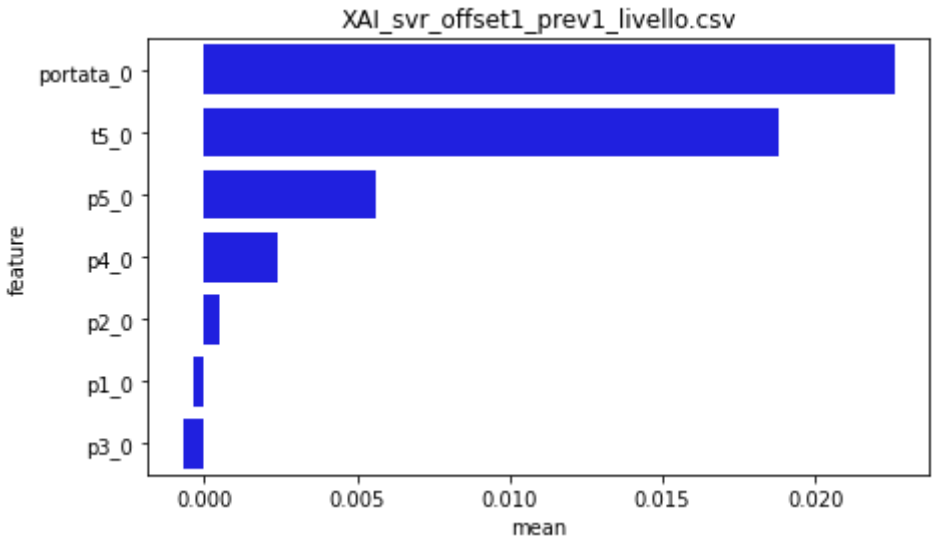
*MAE per ogni combinazione di prev e offset, sulla variabile di target **livello,** con il modello SVR.*



*MAE per ogni combinazione di prev e offset, sulla variabile di target **portata**, con il modello FNN.*

6. Considerazioni e sviluppi futuri

Per prima cosa, il dataset iniziale poteva essere migliore e può essere migliorato: il fatto che ci sia la misura della temperatura per solamente una zona ha pesato sull'efficacia dei modelli, data la valenza oggettiva di quel dato come si può notare dall'ispezione delle permutation importances [4] effettuate sul modello SVR utilizzato per predire il target livello.



Inoltre, la probabile vicinanza delle zone nelle quali sono collocati i sensori sicuramente influenza l'informazione che viene raccolta, come osservato sia dalla matrice di correlazione che dal metodo di explainability, rendendo teoricamente in alcune occasioni inutile l'uso di tutte le variabili (i.e. non piove in nessuna zona) ma praticamente utile ai fini della previsione.

Un'analisi più approfondita dei rischi principali di sovrastima e sottostima, a cura di un esperto di dominio, per quanto riguarda il target livello, può essere utile al fine di selezionare il modello più appropriato.

Alcune analisi che possono essere portate avanti in futuro riguardano ad esempio il numero di prev scelti per la realizzazione dei modelli. Come si è notato, è chiaro che l'utilizzo di più dati porta a risultati migliori ma fino ad un certo punto. Durante la nostra ricerca sono stati individuati momenti in cui il modello convergeva, e dunque, l'utilizzo di moli maggiori di dati si dimostrava inutile ai fini del miglioramento delle performance. Potrebbe essere interessante protrarre questa analisi anche nei casi in cui, da parte nostra, non sono state osservate convergenze.

La previsione della variabile target portata si è rivelata essere piena di insidie. Data la sua natura sparsa, ovvero è spesso costante con sporadici picchi, può essere paragonata ad un problema di classificazione con classi sbilanciate. Per cercare di risolvere abbiamo provato

varie tecniche, tra cui l'applicazione del logaritmo, che avrebbe dovuto avvicinare i valori e quindi rendere più semplice la predizione, ma con scarso successo.

Infine, ultimo ma non per importanza, il vincolo impostoci dall'utilizzo di reti neurali, da un lato molto potenti e di prospettiva ad eccellenti risultati, dall'altro estremamente avide di risorse al punto da rendere temporalmente impossibile la costruzione di troppe istanze. Sarebbe stato interessante, infatti, effettuare più prove con i modelli LSTM, magari provando più svariate architetture.

7. Bibliografia

[1] scikit-learn. 2022. *sklearn.svm.SVR*. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html>

[2] scikit-learn. 2022. *1.6. Nearest Neighbors*. [online] Available at: <https://scikit-learn.org/stable/modules/neighbors.html>

[3] Team, K., 2022. *Keras: the Python deep learning API*. [online] Keras.io. Available at: <https://keras.io/>

[4] scikit-learn. 2022. *sklearn.inspection.permutation_importance*. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.inspection.permutation_importance.html>