Alberi di regressione e analisi di serie temporali: due approcci a confronto per la predizione del prezzo dell'energia in Spagna

Gravina Greta, Pasinato Alessio, Rocchi Niccolò, Scatassi Marco
Università degli Studi di Milano Bicocca, A.A 2021/2022

Abstract

In uno studio pubblicato nel 2019 ([1]) si sottolinea come la predizione in ambito energetico sia uno dei contributi più importanti del Machine Learning. Lo scopo della ricerca è, infatti, quello di affrontare la previsione del prezzo orario dell'energia spagnola con due approcci distinti. Il primo, la regressione basata su alberi decisionali, sfrutta le informazioni riguardo il carico energetico prodotto da varie fonti e, con l'ipotesi di causalità di queste sul prezzo, tenta di inferirlo. Il secondo, invece, sfrutta unicamente le caratteristiche della serie temporale associata all'andamento del prezzo e conduce un'analisi di predizione nell'immediato futuro. Per ognuno dei due approcci sono messi a confronto due algoritmi: Random Forest e XGBoost per il primo, ARIMA e LSTM per il secondo. I risultati complessivi, riassunti in termini di indici statistici, sono infine confrontati. I primi due modelli differiscono significativamente, mentre l'uso di ARIMA e LSTM dipende dalla granularità che si vuole ottenere e dal numero di giorni futuri che si vogliono predire.

Keywords— Machine Learning, Regressione, Serie temporali, Energia, Prezzo

Indice

1 Introduzione

Il cambiamento climatico è una delle più grandi sfide che l'umanità sta affrontando. Tutto ciò che riguarda l'energia prodotta o consumata e il relativo prezzo è impattante su ogni decisione che riguarda il creare nuove infrastrutture economiche adatte ad affrontare i problemi odierni, ma anche a favorire la transizione verso una nuova ecologia energetica. Il contributo che possiamo dare noi studenti di Machine Learning è quello di cercare di applicare metodi predittivi in questo ambito e cercare così di costruire modelli che possano essere di aiuto a chi, un domani, dovesse prendere decisioni d'impatto sul futuro. In particolare, analizzando i dati sulla produzione oraria di energia in Spagna dal 2015 al 2018, abbiamo cercato di predirne il relativo prezzo con due approcci distinti. Il primo cerca di inferire il costo dell'energia a seconda delle fonti utilizzate per produrla; questo può essere utile per chi, prevedendo di produrre una certa quantità di energia, necessita di sapere il relativo prezzo e quindi di basare su questo decisioni politiche, aziendali o strategie economiche di prezzaggio. Il secondo, a differenza del precedente, tenta, a partire dall'andamento oscillatorio del

prezzo nel passato, di descrivere come questo possa essere in un futuro vicino. L'approccio *future forecast* è di estrema importanza sulla descrizione dello scenario delle ore o dei giorni successivi, e anche su questi risultati si possono basare importanti decisioni economiche. I modelli sono infine confrontati sulla base di diversi indici statistici.

2 Descrizione del data set

Per condurre la nostra analisi abbiamo utilizzato il dataset *"Hourly energy demand generation and weather. Electrical demand, generation by type, prices and weather in Spain"* ([3]) messo a disposizione sulla piattaforma *Kaggle* ([2]) nel 2020. Il dataset contiene lo storico dei dati relativi al consumo e alla produzione di energia elettrica di quattro anni (dal 2015 al 2018) ed è stato rilevato dal *Transmission Service Operator (TSO)*. I dati relativi ai prezzi sono stati rilevati dal TSO spagnolo *Red Electric España*. La raccolta dati si compone di 27 attributi numerici e 2 attributi stringa (Time, Forecast wind offshore eday ahead):

1. Time: orario espresso in Anno:mese:giorno:ora
2. Generation fossil gas: produzione di gas fossile
3. Generation fossil hard coal: produzione di carbon fossile
4. Generation fossil oil: produzione di olio fossile
5. Generation fossil oil shale: produzione di scisto bituminoso fossile
6. Generation fossil peat: produzione di torba fossile
7. Generation geothermal: produzione geotermica
8. Generation hydro pumped storage aggregated: produzione di stoccaggio idroelettrico aggregato
9. Generation hydro pumped storage consumption: produzione di consumo di stoccaggio idroelettrico
10. Generation hydro run-of-river and poundage: produzione idroelettrica a cordo d'acqua e carico
11. Generation hydro water reservoir: produzione idroelettrica di riserva d'acqua
12. Generation marine: produzione marina
13. Generation nuclear: produzione nucleare
14. Generation other: altra produzione
15. Generation other renewable: produzione di altra energia rinnovabile
16. Generation solar: produzione solare
17. Generation waste: produzione di rifiuti
18. Generation wind offshore: produzione eolica lontano dalla costa
19. Generation wind onshore: produzione eolica a terra
20. Forecast solar day ahead: previsione della produzione di energia solare del giorno successivo
21. Forecast wind offshore eday ahead: previsione dell'energia eolica lontano dalla costa per il giorno successivo
22. Forecast wind onshore day ahead: previsione dell'energia eolica a terra del giorno prima
23. Total load forecast: previsione del carico totale
24. Total load actual: carico totale effettivo
25. Price day ahead: prezzo del giorno prima
26. Price actual: prezzo effettivo orario

Durante l'esplorazione del dataset si è deciso di escludere dall'analisi le variabili contenenti solo valori nulli (Generation fossil coal derived gas, Generation fossil oil shale, Generation fossil peat, Generation geothermal, Generation marine, Generation wind offshore) e due variabili contenenti solo missing values (Generation hydro pumped storage aggregated, Forecast wind offshore eday ahead). Per rispondere alla prima domanda di ricerca con cui si è cercato di inferire il costo dell'energia a seconda delle fonti da cui è prodotta, sono state utilizzate tutte le variabili relative alla produzione di energia e la variabile prezzo effettivo (target). Invece, per rispondere alla seconda domanda di ricerca che predice il costo futuro dell'energia studiando l'oscillazione del prezzo nel passato, sono state considerate solamente le variabili tempo e prezzo effettivo (target). Inoltre, per l'implementazione dei quattro modelli di predizione Random Forest, XGBoost, ARIMA e LSTM, sono state condotte fasi di preprocessing diverse in base alle caratteristiche dell'algoritmo preso in esame (Sezione 4).

3 Modelli e misure di performance

3.1 Modelli

Individuati gli algoritmi di apprendimento più adatti al nostro scopo, si è proseguito costruendo i relativi modelli attraverso un workflow sulla piattaforma *Knime*. I modelli implementati sono:

1. *Random Forest*: è un algoritmo di Machine Learning supervisionato che permette di risolvere problemi di classificazione e di regressione. Per quanto riguarda la seconda tipologia di task, questo opera costruendo molti alberi decisionali durante la fase di training fornendo in uscita la media di tutte le previsioni generate dagli alberi in modo da determinare una stima ottimale della variabile target. Random Forest Regression è un modello potente ed accurato grazie al suo approccio *wisdom of the crowds* (letteralmente *saggezza della folla*) e riesce solitamente a scalare bene quando vengono aggiunte nuove variabili al data set. E' interpretabile e facile da usare.

2. *XGBoost*: è un algoritmo di Machine Learning basato su alberi decisionali, noto per la sua efficienza e scalabilità, implementato per mezzo di *Knime XGBoost Integration*[1].

3. *AutoRegressive Integrated Moving Average*: i modelli ARIMA sono la generalizzazione di quelli ARMA (*AutoRegressive Moving Average*). Si basano su valori passati di una serie temporale con ipotesi di stazionarietà e non-stagionalità, per riuscire a predire valori futuri.

4. *Long short-term memory*: LSTM è un tipo particolare di RNN (*Recurrent Neural Network*) capace di memorizzare un grande numero di input precedenti per cui adatto a problemi di analisi di serie temporali e quindi a task di predizione futura.

3.2 Misure di performance

Per confrontare i modelli sono stati utilizzati quattro indicatori statistici: R^2, MAE, RMSE e MAPE. Definiamo innanzitutto $\hat{y}_i$ come la previsione del punto dato y_i al i-esimo tempo t e $\bar{y}$ la media dei dati nell'arco di tempo considerato. Il primo indice è calcolato secondo la seguente definizione ed indica la varianza totale spiegata dal modello:

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Tale indice è utile nell'ambito della regressione lineare semplice, ed esiste una versione *aggiustata* nel caso di molteplicità dei regressori. Tuttavia si è deciso di tralasciare tale misura poichè dagli esperimenti è risultata non significativamente differente dal precedente. Come suggerisce la letteratura ([7]), tale indice non è da considerarsi nell'approccio di previsione futura della serie temporale, ma solo nell'ambito di regressione, in quanto ci si concentra sull'accuratezza della previsione futura, e non sulla bontà del modello nel passato.
Gli indici MAE e RMSE, invece, sono misure che riguardano la media degli errori. Questi non possono essere utilizzati per confrontare predizioni di variabili con diverse unità di misura poichè condividono la stessa della variabile target. Indicando $e_i \coloneqq y_i - \hat{y}_i$ si definiscono

$$\text{MAE} = \text{mean}(|e_i|)$$

$$\text{RMSE} = \sqrt{\text{mean}(e_i^2)}$$

Si osservi che questi indici dipendono dalla scala del fenomeno e non tengono conto del rapporto dell'errore con il valore osservato. Per questo l'indice più robusto nel nostro caso è il MAPE, ed è sinonimo di bontà per ogni modello considerato. L'indice MAPE calcola la media degli errori relativi, ovvero:

$$\text{MAPE} = \text{mean}\left(\frac{|e_i|}{|y_i|}\right)$$

Il difetto di questo è che risulta indefinito nel caso in cui la i-esima osservazione sia nulla. Si noti che queste ultime misure di bontà, essendo medie, dipendono dal numero di osservazioni considerate, ovvero variano a seconda del denominatore preso in esame. Per questo motivo si è deciso di adottare, per Random Forest e XGBoost, una Cross Validation e un Holdout che considerino la stessa partizione del data set.
Con riferimento a [4], la seguente tabella indica l'accuratezza predittiva del modello in funzione dell'indice MAPE.

[1] https://hub.knime.com/knime/extensions/org.knime.features.xgboost/latest

Valore	Accuratezza predittiva
MAPE ≤ 0.1	Alta
$0.1 <$ MAPE ≤ 0.2	Buona
$0.2 <$ MAPE ≤ 0.5	Ragionabile
MAPE > 0.5	Bassa

Tabella 1: Accuratezza predittiva in funzione del MAPE

4 Analisi e risultati

4.1 Prima domanda di ricerca

"E' possibile predire il prezzo dell'energia elettrica considerando la produzione derivante dalle differenti fonti di energia?". Al fine di dare una risposta abbiamo sfruttato gli algoritmi Random Forest e XGBoost utilizzando come variabile target il prezzo effettivo e come variabili esplicative le diverse produzioni di energia. Sono state utilizzate le tecniche di Holdout e Cross Validation per testare le loro performance utilizzando per entrambe lo stesso partizionamento.
Per quanto concerne l'Holdout, il dataset è stato suddiviso in due partizioni composte da record estratti in maniera casuale dall'insieme originario: la prima contenente il 67% dei record totali, la seconda il 33%. Entrambi i modelli sono stati poi addestrati con i relativi algoritmi sulla prima partizione, ovvero il training set, e testati sulla seconda partizione, il test set. Per il processo di Cross Validation abbiamo deciso di utilizzare una 10-fold Cross Validation: in ciascuna iterazione sono stati registrati i parametri relativi alla performance.

Random Forest Il modello Random Forest è stato costruito utilizzando il nodo Knime *Random Forest Learner (Regression)* impostato con 100 alberi di regressione da cui imparare con ciascuno un numero minimo di 7 nodi figli che permette di addestrare il modello a partire dai dati di training, e il nodo *Random Forest Predictor (Regression)* per testare il modello sui dati di test set. Si è deciso di costruire il modello utilizzando 100 alberi decisionali come parametro di profondità poichè il data set è composto da un numero consistente di record e le performance del modello miglioravano man mano che si aumentava il numero di alberi fino a raggiungere una situazione di stabilità con 100 alberi. Dopo le prime versioni del modello valutate tramite Holdout e Cross Validation ci si è chiesti se i parametri delle performance sul test set potessero essere migliorate. Per questo motivo sono stati corretti i valori anomali del data set sostituendo a questi i valori più vicini.

XGBoost Diversamente dall'algoritmo Random Forest, XGBoost non risulta essere influenzato negativamente dalle osservazioni contenenti valori anomali, anzi, rimuovendoli si nota un leggero calo nell'efficienza predittiva; si è deciso quindi di non eliminare tali istanze. Per quanto riguarda la performance, questa è stata migliorata modificando due parametri: il primo, *maximum depth*, indica la profondità massima raggiungibile da un albero, ed è stato incrementato di un'unità, passando da 6 a 7, rendendo la struttura del modello leggermente più complessa. Si è però evitato di incrementare ulteriormente questo valore, perché col crescere di esso aumentano anche le probabilità di overfitting sul training set. Il secondo parametro che ha contribuito a migliorare le prestazioni è *minimum child weight*, il quale indica il numero minimo di istanze che devono essere presenti in ciascun nodo, ed è stato incrementato da 1 a 2 ([6]).

Le seguenti immagini mostrano, per ognuno dei due algoritmi, un particolare del comportamento orario del prezzo e della sua predizione tramite Holdout.

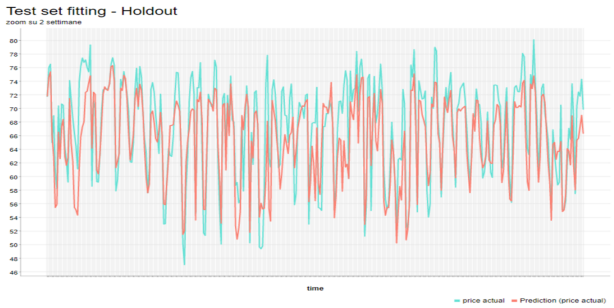

Figura 1: Test set fitting - Holdout (Random Forest)

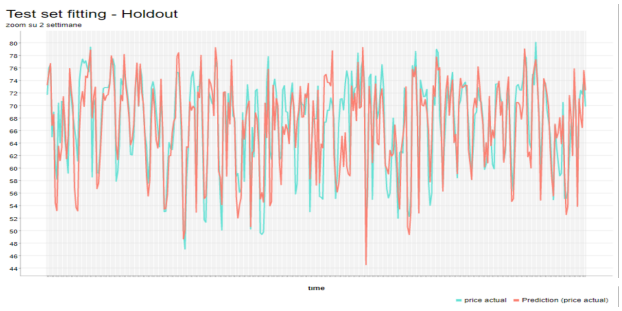

Figura 2: Test set fitting - Holdout (XGBoost)

Per quanto riguarda l'Holdout, sembra che XGBoost sia il modello più efficiente nel predire il prezzo dell'elettricità. Il modello infatti riesce ad apprendere meglio rispetto alla Random Forest. Sui dati di training presenta un R^2 pari a 0.93, più alto di quello della Random Forest (0.86) e MAE,

RMSE e MAPE più bassi. In aggiunta, XGBoost ottiene risultati migliori anche sui dati di test con un R^2 pari a 0.81, un MAE di 4.50, RMSE di 6.23 e MAPE di 0.1 (unico parametro che condivide con la Random Forest).

Anche per quanto riguarda la 10-folds Cross Validation il modello XGBoost è significativamente migliore rispetto agli indici MAE, RMSE e MAPE; indice di questo fatto è la differenza tra le mediane e la non sovrapposizione dei box plot.

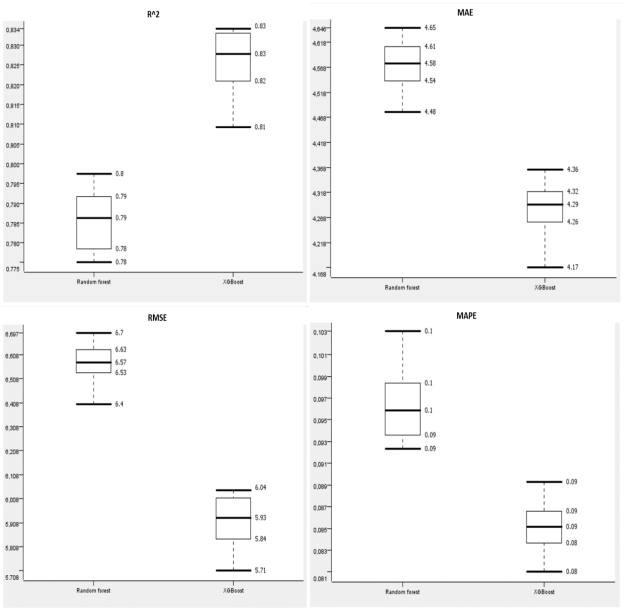

Figura 3: Confronto tra Random Forest e XGBoost tramite 10-folds Cross Validation

In conclusione, dalle analisi effettuate, il modello XGBoost è il modello più efficiente per predire il prezzo dell'elettricità.

4.2 Seconda domanda di ricerca

Come secondo obiettivo ci siamo posti quello di predire il prezzo dell'energia elettrica il giorno successivo, considerando l'andamento della serie temporale associata. La fase di preprocessing tra ARIMA e LSTM è simile, con qualche differenza. In entrambi i casi, dopo avere trasformato il tempo da tipo stringa a tipo timestamp si è esclusa ogni variabile diversa da time e price actual. Il dataset presentava inoltre orari duplicati (osservazioni diverse con la stessa variabile tempo): in particolare, a otto date erano associate due osservazioni distinte in relazione ai restanti attributi tra cui il prezzo. Non conoscendone il vero valore si è deciso di fare una media dei prezzi corrispondenti a una stessa data, e sono state infine riordinate le osservazioni in base all'attributo tempo in ordine crescente. Nel primo modello si è deciso di considerare una granularità giornaliera, in cui il prezzo è sostituito dalla media dei prezzi del giorno, ed eventuali valori mancanti sono stati sostituiti con il valore zero; questo permette una maggiore efficienza e robustezza del modello. Per quanto riguarda LSTM, invece, la granularità del dataset è mantenuta oraria e i valori mancanti sono estratti per interpolazione lineare tra il successivo e il precedente valore non nullo del data set. Inoltre la variabile prezzo è stata normalizzata nell'intervallo $[0, 1]$.

4.2.1 ARIMA

Il modello prevede un'ipotesi di stazionarità e non stagionalità. A questo scopo la serie $S = S(t)$ è stata scomposta come segue:

$$S = T + S_1 + S_2 + R$$

dove T è la componente di trend, S_1,S_2 le componenti stagionali e R il residuo. S_1 è calcolata in base alla curva ACF (*Auto-correlation function*) -la quale presenta picchi ogni 7 giorni, ad indicare una autocorrelazione settimanale- mentre S_2 si è rivelata nulla, in quanto l'unica stagionalità riscontrata è la prima. Questa decomposizione permette al modello di funzionare al meglio, siccome la variabile residuo è non stagionale e, come si è verificato graficamente, stazionaria. In aggiunta si è deciso di rimpiazzare i valori anomali, ovvero quelli esterni al box plot della distribuzione, con il valore più vicino.

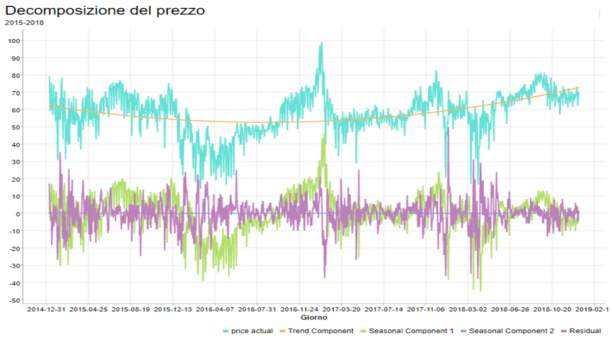

Figura 4: Decomposizione del prezzo

Il modello ARIMA prevede la scelta di tre parametri (p, d, q). Per quanto riguarda la predizione del residuo, riferendosi alla letteratura ([9]), il primo parametro è stato impostato

come il valore di lag nel quale la curva PACF (*Partial AutoCorrelation Function*) presenta il primo cutoff. Il secondo è stato impostato a zero data l'ipotesi di stazionarietà del residuo, mentre l'ultimo è pari a 1, e non maggiore, per non annullare gli effetti del primo. Per quanto riguarda invece la predizione del trend si è deciso di utilizzare un modello ARIMA(0,0,0) siccome la curva è quasi lineare.
Si è dunque proseguito allenando il modello sui dati dal 2015 al 2017. Dopo averne visualizzato la bontà nel descrivere i dati di training, si è testato sui dati di test come segue. Per ogni giorno del 2018 si è predetto il valore residuo e il valore trend. Per quanto riguarda la componente stagionale, siccome l'unica autocorrelazione trovata è stata quella settimanale, ad ogni valore r_i del residuo predetto è stato sommato r_{i-7} ovvero il residuo una settimana prima. A questo punto è stato sommato il trend predetto ed è stato confrontato il valore finale con quello reale. Questo procedimento è stato iterato per ogni giorno del 2018, ogni volta collezionando i dati, per poi calcolarne gli indici statistici. Per il confronto con LSTM si veda 4.2.2.
Si è sperimentato infine come potrebbero cambiare i parametri statistici del procedimento precedente nel caso in cui si predica fino a 5 giorni dopo. Per ogni blocco di 5 giorni si sono registrati gli indici di bontà; ciò che risulta è che la mediana del MAPE arriva a 0.16, ad indicare una robustezza del modello anche nel caso in cui si voglia predire un arco di tempo più ampio. Nonostante, come si vedrà in seguito, il modello LSTM ottiene un indice MAPE nettamente migliore nella predizione del giorno successivo, ARIMA riesce tuttavia a ottenere una predizione abbastanza accurata anche considerando qualche giorno successivo.
L'ultima parte del workflow prevede infatti un *model deployment*: una semplice interfaccia che prevede di inserire un giorno di partenza e un numero n di giorni come input, e che prevede i valori del prezzo fino a n giorni dopo il punto di partenza.

4.2.2 LSTM

LSTM è stato utilizzato per risolvere due compiti principali:

1. Prevedere il prezzo dell'energia nell'ora successiva a una certa data sulla base dei valori assunti dal prezzo nel passato (*In-Sampling*);

2. Prevedere il prezzo dell'energia nelle 24 ore successive a una certa data sulla base dei valori assunti dal prezzo nel passato (*Out-Sampling*).

Per addestrare il modello è innanzitutto necessario definire la rete neurale di apprendimento di input del learner. Quella utilizzata è costituita da:

- Un *layer* di input che accetta vettori contenenti 72 valori passati (ovvero 3 giorni) ;

- Un *hidden layer* con 100 unità LSTM;

- Un *dense layer* di output con un solo neurone e funzione di attivazione *ReLu*.

Si è poi suddiviso il data set in training, validation e test set. Poiché il modello necessita che i valori siano temporalmente ordinati, il training set è stato ottenuto prendendo quasi tutti i valori dei primi tre anni e l'anno restante è stato suddiviso quasi a metà per costruire il validation e il test set. Sono stati poi addestrati tre modelli che differiscono per la funzione di perdita utilizzata (Modello 1: *Pinball* con percentile 0.9, Modello 2: *Pinball* con percentile 0.1, Modello 3: *MSE-Mean Squared Error*). Per rendere i tre modelli confrontabili gli altri parametri, necessari all'apprendimento, sono stati impostati allo stesso modo (50 epoche, dimensione batch di 512 per il training set e di 256 per il validation set, ottimizzatore Adam, learning rate 1E-4, etc.). In particolare, per limitare il fenomeno dell'overfitting, si è posta come condizione di termine dell'addestramento, oltre al raggiungimento del numero massimo di epoche, lo stop nel caso in cui la funzione di perdita tra due epoche successive, rispetto al validation set, subisse un incremento. Le previsioni quantiliche sono utili a comprendere l'attendibilità della predizione fatta attraverso il Modello 3: quello che ci si aspetta è che:

$$p_2(\text{tempo}) \leq p_3(\text{tempo}) \leq p_1(\text{tempo})$$

dove $p_i(\text{tempo})$ indica la previsione fatta dal modello i-esimo in una certo tempo.

Task 1: In-Sampling L'immagine seguente mostra gli andamenti dei valori predetti dai tre modelli e quello reale del prezzo per le prime due settimane della seconda metà del 2018, quindi relativamente ai primi 336 valori del test set:

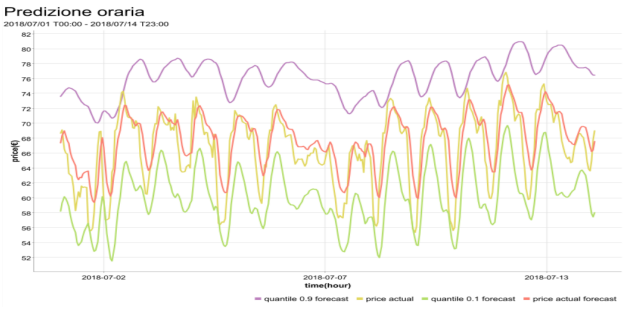

Figura 5: Campione di previsione di LSTM

Da questa si osserva come in effetti il modello M1 generi una curva che sta quasi ovunque al di sopra delle altre, il modello M2 generi una curva che sta quasi ovunque al di sotto delle altre, e il modello M3 generi la curva maggiormente aderente a quella reale.

Task 2: Out-Sampling Si è poi provato a utilizzare il modello 3 per predire le 24 ore successive a una certa data basandosi sulle 72 ore precedenti a questa. L'idea è del tutto analoga a quella usata per il modello ARIMA. I risultati ottenuti sono sintetizzati nel box plot seguente (mediana del MAPE: 0.08):

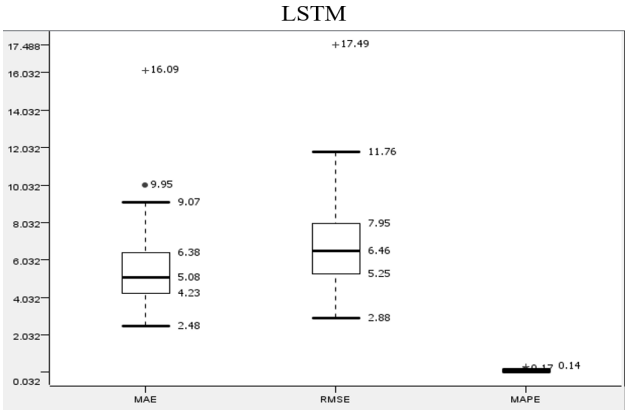

Figura 6: Indici statistici del modello LSTM

Per poter avere poi un confronto col modello ARIMA è stata usata la media delle 24 previsioni orarie, ottenute col modello LSTM, come stima della media giornaliera del prezzo reale. Le statistiche ottenute usando i due modelli sono riportate nella seguente tabella:

	ARIMA	LSTM
MAE	10.59	3.99
RMSE	12.79	4.74
MAPE	0.18	0.06

Tabella 2: Confronto tra ARIMA e LSTM

In particolare, la media delle previsioni sulle 24 ore giornaliere ottenute con LSTM riesce a stimare bene quella reale, seppur l'andamento delle previsioni sulle 24 ore non sia in grado di cogliere le fluttuazioni del prezzo reale che tende ad avere un picco minimo nelle ore iniziali della giornata e ad avere un picco massimo nelle ore serali come riportato nella figura seguente.

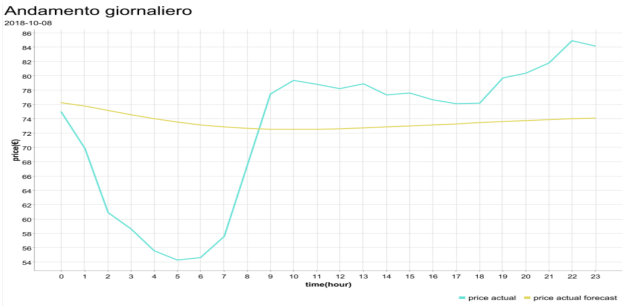

Figura 7: Previsione giornaliera del prezzo

5 Conclusioni

Provare a risolvere un task di regressione come la predizione del costo dell'energia elettrica ci ha permesso di studiare e successivamente implementare modelli predittivi sfruttando algoritmi di Machine Learning supervisionati differenti. Dall'analisi risulta che, per rispondere alla prima domanda di ricerca, XGboost è il miglior metodo predittivo, e può dunque risultare utile per predire il prezzo dell'energia a partire dalle diverse produzioni raccolte negli anni passati; considerando invece la variabile temporale, sia ARIMA che LSTM riescono a fare predizioni accurate sul futuro. Il primo modello, in particolare, prevede il costo dell'energia elettrica media non solo per il giorno successivo, ma fino a 5 giorni dopo con buoni risultati. Il secondo modello riesce ad ottenere una maggiore accuratezza sulla media del giorno successivo e una maggiore granularità della predizione. La scelta di uno di questi ultimi due modelli dipende dalle esigenze specifiche e dalle proprie domande di ricerca.

Riferimenti

[1] *Tackling Climate Change with Machine Learning*, 5 nov. 2019

[2] Kaggle. https://www.kaggle.com/

[3] Data set. https://www.kaggle.com/nicholasjhana/energy-consumption-generation-prices-and-weather?select=energy_dataset.csv

[4] https://www.researchgate.net/figure/MAPE-Value-for-Prediction-Evaluation-36_tbl2_318055885

[5] https://www.keboola.com/blog/random-forest-regression

[6] https://xgboost.readthedocs.io/en/stable/parameter.html

[7] https://www.knime.com/blog/building-a-time-series-analysis-application

[8] https://www.knime.com/blog/time-series-analysis-with-components

[9] https://people.duke.edu/ rnau/arimrule.htm

[10] https://towardsdatascience.com/mad-over-mape-a86a8d831447

[11] *Codeless Deep Learning with KNIME*, K.Melcher, R.Silipo, 27 nov. 2020

[12] https://towardsdatascience.com/probabilistic-forecasts-pinball-loss-function-baf86a5a14d0

[13] https://colah.github.io/posts/2015-08-Understanding-LSTMs/