POLITECNICO DI TORINO

Laurea Magistrale in Ingegneria Energetica

Previsione del Prezzo Unico Nazionale in un Contesto di Scenari Prestabiliti:

Approccio Statistitico e Machine Learning per la Previsione del Mercato Elettrico Italiano

Relatori:
Prof. Maurizio Repetto
Prof. Paolo Lazzeroni

Candidato:
Manuel Gallo - 305421

DENERG
Politecnico di Torino
Italia
Marzo 2024

Abstract

L'evoluzione del mercato elettrico italiano, dalla sua nascita come monopolio statale fino alla completa liberalizzazione, ha segnato una trasformazione fondamentale non solo per l'industria energetica del paese ma anche per il modo in cui consumatori e aziende interagiscono con questo bene primario.

Nel contesto di questo cambiamento, la necessità di sviluppare strumenti e metodologie avanzate per la previsione del prezzo dell'energia elettrica è diventata sempre più pressante. La ricerca accademica ha esplorato con rigore questo ambito, adottando una varietà di metodologie che vanno dagli approcci statistici convenzionali fino alle più sofisticate tecniche di machine learning, evidenziando l'importanza vitale di effettuare previsioni accurate per informare le scelte degli attori del mercato.

I modelli statistici, come l'Autoregressive Integrated Moving Average (ARIMA) e gli Ordinary Least Squares (OLS), hanno lungamente dominato il campo della previsione, fornendo interpretazioni basate su principi statistici consolidati. Tuttavia, l'avvento del machine learning ha introdotto un cambio di paradigma con l'adozione di modelli avanzati e non lineari, capaci di decifrare schemi complessi in grandi volumi di dati. Tecnologie quali il *Deep Learning*, le *Support Vector Machines (SVM)* e i metodi *Ensemble (modelli di insieme)* hanno riscosso un crescente interesse, offrendo nuove prospettive e sfide nel miglioramento delle previsioni energetiche.

L'obiettivo principale dell'elaborato è duplice: da un lato, si intende fornire una comprensione dettagliata del funzionamento del mercato elettrico italiano, mettendo in luce i cambiamenti storici e i meccanismi attuali che ne regolano le dinamiche; dall'altro, valutare criticamente diverse tecniche di previsione dei prezzi dell'energia, dall'approccio semplicistico dei modelli auto-regressivi, fino alle tecniche più complesse di intelligenza artificiale dei modelli ensemble, al fine di fornire un contributo metodologico che permetta di analizzare il Prezzo Unico

Nazionale (PUN) italiano su un orizzonte temporale decennale, tenendo conto delle potenzali variazioni delle condizioni a contorno.

In particolare, verranno esaminati il modello *SARIMA (Seasonal Autoregressive Integrated Moving Average)*, la *Regressione Lineare Multipla* e il *XGBoost (Extreme Gradient Boosting)*, confrontando le prestazioni predittive dei modelli statistici e autoregressivi con quelle delle avanzate tecniche di machine learning basate su modelli di insieme.

Una volta individuato il modello migliore in termini di adattabilità ai dati, verrà eseguita un'analisi previsionale tenendo conto di due potenziali scenari futuri: uno caratterizzato da un significativo aumento del prezzo del *gas naturale*, potenzialmente scaturito da contesti geopolitici critici, e l'altro da un aumento del *fabbisogno elettrico nazionale*, in linea con gli obiettivi europei di riduzione delle emissioni del 55% entro il 2030.

Le conclusioni, infine, mirano a delineare potenzialità e limiti delle metodologie esaminate, offrendo un'analisi critica basata sull'accuratezza delle previsioni e sulla loro applicabilità pratica nel contesto del mercato elettrico italiano, evidenziando così le direzioni future per la ricerca e lo sviluppo nel settore.

Indice

Elenco delle tabelle

Elenco delle figure

Acronimi

PUN
: Prezzo unico nazionale

MGP
: Mercato del giorno prima

MI
: Mercato infra-giornaliero

MSD
: Mercato dei servizi di dispacciamento

MTE
: Mercato a termine

IPEX
: Italian Power Exchange

TSO
: Transmission System Operator

ISO
: Independent System Operator

GME
: Gestore mercati energetici

GSE
: Gestore Sistemi Energetici

RTN

Rete di trasmissione nazionale

PNIEC

Piano Nazionale Integrato per l'Energia e il Clima

GNL

Gas naturale liquefatto

DEM

Fabbisogno di energia elettrica

SOLAR

Energia generata da impianti fotovoltaici

WIND

Energia generata da impianti eolici

SARIMA

Seasonal Auto-Regressive Integrated Mean Average

OLS

Ordinary Lean Square

XGB

Extreme Gradient Boosting

MAE

Mean Absolute Error

RMSE

Root Mean Square Error

R2

Coefficiente di determinazione

Capitolo 1

Contesto Energetico e Normativo

1.1 Politica Energetica Europea

La politica energetica ha sempre rivestito un ruolo centrale nelle prerogative nazionali, ma è stata anche un pilastro fondamentale per l'integrazione europea fin dalle sue origini. Basti pensare che già nel 1951, il Trattato della *Comunità Europea del Carbone e dell'Acciaio* (*CECA*) si poneva l'obiettivo di regolare produzione e distribuzione del carbone ponendo le basi per la *sicurezza energetica* come elemento chiave dell'integrazione europea.

Nel settembre del 1974, con una risoluzione del Consiglio, l'Europa ha iniziato a delineare un quadro comunitario di politica energetica segnando un primo passo verso una strategia energetica condivisa. L'adesione alla *Convenzione Quadro delle Nazioni Unite sui Cambiamenti Climatici* (conosciuta anche come gli *Accordi di Rio*) e la successiva ratifica del *Protocollo di Kyoto* hanno costituito le fondamenta per un'agenda globale mirata alla riduzione delle emissioni di gas serra. Questi impegni sono stati rafforzati dalle *Conferenze delle Parti* (*COP*), incontri annuali dove i paesi aderenti valutano i progressi e si accordano sugli obiettivi futuri.

L'Europa ha agito con determinazione anticipando le altre potenze mondiali nell'adottare misure per il raggiungimento degli obiettivi stabiliti. Tra queste l'introduzione del sistema di scambio delle quote di emissione (*Emission Trading System*), ossia il primo schema globale nel suo genere volto a ridurre le emissioni di CO_2. Inoltre, adottando l'ambizioso obiettivo "20-20-20" entro il 2020 con il "Pacchetto clima-energia" lanciato nel 2009, si pose l'obiettivo di favorire la decarbonizzazione del sistema energetico europeo, la quale sarebbe stata possibile riducendo del 20% le emissioni di gas serra rispetto ai livelli del 1990, raggiungendo una quota del 20% di energia da fonti rinnovabili sul totale dei consumi energetici e riducendo del 20% i consumi energetici, basati quest'ultimi su previsioni di scenario prestabilite.

Basandosi sui risultati iniziali delle politiche introdotte dal 2009, nel gennaio 2014 la Commissione ha proposto i primi obiettivi climatici ed energetici per il 2030, con l'obiettivo di consolidare una posizione comune europea ben prima della Conferenza sul Cambiamento Climatico di Parigi (COP 21) del Dicembre 2015. Questo evento rappresentava un'opportunità cruciale per influenzare il regime internazionale post-Kyoto, con l'UE che puntava a esercitare pressione sugli altri attori globali attraverso un impegno credibile e ambizioso.
Tuttavia, le divergenze tra le posizioni degli Stati membri e del Parlamento Europeo hanno messo in luce le preoccupazioni relative ai costi della transizione energetica e alla possibile perdita di sovranità nelle politiche nazionali [1].

La strategia 20-20-20 venne dunque seguita dalla *Strategia 2030*, annunciata alla fine del 2018, la quale si pone l'obiettivo di superare gli obiettivi della strategia precedente per avvicinarsi agli scopi della *Roadmap 2050*, che ambisce a una decarbonizzazione quasi totale del sistema energetico, con una riduzione delle emissioni di gas serra dell'$80 \div 95\%$.

In particolare, gli obiettivi UE 2030 Clima-Energia includono:

- Un incremento al 40% nella riduzione delle emissioni di gas a effetto serra rispetto ai livelli del 1990, attraverso una diminuzione del 43% delle emissioni per i settori coperti dal sistema di scambio di quote di emissione (ETS) rispetto al 2005, e una riduzione del 30% per le emissioni dei settori non inclusi nell'ETS, sempre rispetto al 2005;

- Un rialzo degli obiettivi iniziali per la percentuale di energia consumata prodotta da fonti rinnovabili e per l'efficienza energetica, che erano stati fissati al 27% e successivamente aggiornati al 32% e al 32,5% nel 2018.

La Strategia 2030 prevede una maggiore flessibilità per i governi nazionali, che dovrebbe essere compensata dalla creazione di un quadro di governance europeo tale che renda le politiche attuate dagli Stati membri il più efficaci e trasparenti possibile nel raggiungimento degli obiettivi. In questo contesto, è stato previsto che gli Stati membri elaborassero un Piano Nazionale Integrato per l'Energia e il Clima (*PNIEC*) riferito al periodo 2021-2030 in forma definitiva entro il 2019, e si impegnassero in strategie nazionali a lungo termine [2].

1.2 Politica Energetica Italiana: PNIEC e PIANO 2030

L'Italia si è distinta come uno dei paesi membri più proattivi e cooperativi nell'elaborazione e attuazione delle strategie energetiche dell'Unione Europea.
Nel contesto del PNIEC, l'Italia ha posto particolare enfasi sul ruolo del gas naturale, mirando ad espandere significativamente la capacità di importazione del gas naturale liquefatto (GNL).

Gli obiettivi strategici nazionali definiti dall'Italia sono i seguenti:

- **Decarbonizzazione e Sviluppo delle Energie Rinnovabili:** l'obiettivo di eliminare l'uso del carbone nelle centrali elettriche si inserisce in una strategia più ampia che non deriva direttamente dagli accordi europei ma che richiede lo sviluppo di infrastrutture alternative per sostituire l'energia prodotta dal carbone e mantenere stabile il sistema elettrico.
  Sul fronte dell'energia rinnovabile, l'obiettivo è stato stabilito considerando l'obbligo di contribuire agli obiettivi europei, di aumentare la quota di energia rinnovabile consumata e la necessità di limitare l'uso del suolo; ciò ha portato a fissare una quota del 30% di energia rinnovabile sui consumi totali entro il 2030. L'obiettivo richiederà una significativa espansione dell'eolico e del fotovoltaico, con una media annuale di installazione di circa 3200 MW per l'eolico e 3800 MW per il fotovoltaico dal 2019 al 2030. Sarà inoltre necessario sviluppare infrastrutture di accumulo e incrementare l'uso di energia rinnovabile per riscaldamento, raffrescamento e nei trasporti.

- **Efficienza Energetica:** l'Italia punta a superare gli obiettivi europei mirando a una riduzione del 43% nel fabbisogno di energia primaria entro il 2030, rispetto alle proiezioni del 2007. Inoltre, si prevede di ridurre i consumi finali di energia di circa lo 0,8% annuo rispetto alla media del triennio 2016-2018, un traguardo che richiederà sforzi considerevoli anche nei settori più impegnativi come edilizia e trasporti.
  L'elettrificazione dei trasporti gioca un ruolo chiave, con l'obiettivo di 1,6 milioni di auto elettriche e 4,5 milioni di ibride entro il 2030.

- **Sicurezza Energetica:** l'obiettivo è migliorare la sicurezza degli approvvigionamenti attraverso la diversificazione delle fonti e l'aumento dell'efficienza energetica e delle rinnovabili. Si prevede inoltre di sfruttare il potenziale del biometano e di integrare il sistema gas con quello elettrico.

- **Mercato Interno:** l'Italia mira a incrementare la flessibilità del sistema elettrico e ad evolvere le regole del mercato per facilitare l'integrazione delle rinnovabili. Si punta anche a un maggior coinvolgimento dei consumatori come "prosumer" e alla tutela dei consumatori, con un occhio di riguardo alla lotta contro la povertà energetica.

Questi obiettivi ambiziosi delineano il percorso dell'Italia verso un sistema energetico più sostenibile, efficiente e sicuro, in linea con gli impegni europei e internazionali per il 2030 e oltre.

Capitolo 2

Il Mercato Elettrico e il Prezzo Unico Nazionale

2.1 Genesi del Mercato Elettrico

Il settore elettrico italiano fonda le sue radici sull'intraprendenza di imprenditori locali che, identificando opportunità nelle aree urbane, hanno avviato l'attività elettrica adottando un modello di integrazione verticale.
Inizialmente, ciascun operatore monopolista locale gestiva l'intero processo: dalla generazione alla vendita dell'energia elettrica. Con il progresso tecnologico e l'espansione industriale, questi operatori estesero le loro reti evitando sovrapposizioni territoriali per ottimizzare costi e benefici. Questo portò a un sistema in cui l'energia elettrica, divenuta un bene di pubblico interesse, era controllata da pochi soggetti, lasciando alcune aree prive di servizio per mancanza di interesse economico.

Dopo la Seconda Guerra Mondiale, l'Europa visse una svolta politica e ideologica che favorì lo sviluppo economico e l'opposizione ai monopoli, clima che portò alla *Nazionalizzazione* del settore elettrico in alcuni paesi membri. In Italia, la questione della nazionalizzazione fu risolta il 6 dicembre 1962 con la creazione dell'*ENEL S.p.A.* (Ente Nazionale per l'Energia Elettrica), ispirandosi al modello dell'*Électricité de France*. L'ENEL unificò le imprese elettriche private sotto un unico ente monopolizzando la produzione, trasmissione e distribuzione dell'energia, con l'intento di ottimizzare la gestione e l'infrastruttura della rete elettrica nazionale. Nel tempo, l'ENEL si adattò alle esigenze del mercato diversificando la produzione

e modernizzando gli impianti.

Nel 1999, tuttavia, il *"Decreto Bersani"* sanciva ufficialmente l'inizio della *Liberalizzazione* del mercato elettrico italiano. Tale legislazione introdusse la separazione tra la gestione delle reti e le attività di produzione e vendita, aprendo la generazione di energia alla concorrenza e limitando la quota di mercato dell'ENEL favorendo l'ingresso di nuovi operatori. Il Decreto Bersani, dunque, segnò l'inizio di una serie di riforme che hanno progressivamente diversificato il mercato elettrico italiano.

A seguito dell'avvio del processo di liberalizzazione, il settore energetico italiano si strutturò attorno a quattro pilastri fondamentali: *produzione*, *trasmissione*, *distribuzione* e *commercializzazione e vendita*.

Al termine del 2019, come riportato da *TERNA* (Figura 2.1), il panorama della generazione di energia in Italia ha visto una sostanziale stabilità nella produzione termoelettrica rispetto all'anno precedente, mentre si è assistito a un significativo aumento nella produzione da fonti rinnovabili, che ha raggiunto il 46,5% della capacità installata totale nel paese. Questa tendenza ascendente nelle fonti rinnovabili è prevista proseguire nei prossimi anni, supportando la transizione verso un sistema energetico più sostenibile e meno dipendente dalle fonti fossili. [3]

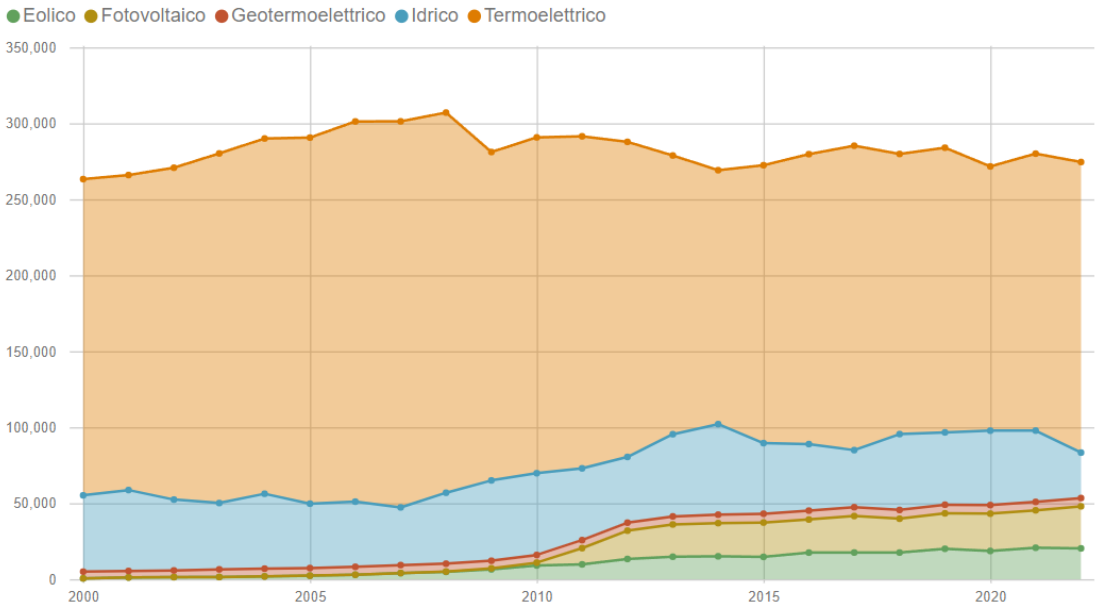

Figura 2.1: Produzione netta di energia elettrica per fonte [GWh] (Fonte:TERNA).

TERNA gioca un ruolo chiave in questa transizione, operando come l'operatore di sistema di trasmissione (*TSO*) e l'operatore di sistema indipendente (*ISO*) in Italia, sotto un regime di monopolio naturale, ossia garantito da una concessione statale. Le sue responsabilità includono la *pianificazione*, lo *sviluppo* e la *manutenzione* della *Rete di Trasmissione Nazionale* (*RTN*), oltre alla *gestione* del flusso di energia elettrica attraverso la rete. TERNA assicura l'equilibrio costante tra domanda e offerta di energia, ma non si occupa direttamente di distribuzione e vendita, attività che sono invece affidate per il 90% a *E-Distribuzione*. Quest'ultima, operativa dal 1° ottobre 1999, gestisce gli aspetti relativi alle infrastrutture di distribuzione comprese cabine, linee di media e bassa tensione e dispositivi di misurazione.

Il *Gestore dei Servizi Elettrici* (*GSE*), controllato interamente dal Ministero dell'Economia e delle Finanze, è un altro ente cruciale nel panorama energetico nazionale, con il compito di incentivare lo sviluppo delle fonti energetiche rinnovabili in Italia. Esso promuove, inoltre, l'efficienza energetica e contribuisce a sensibilizzare sull'uso responsabile dell'energia.

2.2 La Struttura della Borsa Elettrica (IPEX)

Nel contesto del Mercato Elettrico, la *Borsa Elettrica Italiana*, conosciuta come *Italian Power Exchange* (IPEX), istituita nel 2004, offre una piattaforma per la negoziazione di energia mettendo in contatto produttori, venditori e acquirenti. Gestita dal *Gestore dei Mercati Energetici* (*GME*), un organismo che mira a garantire la trasparenza e la competitività del mercato energetico all'ingrosso, la Borsa Elettrica si avvale di un sistema telematico sicuro per facilitare scambi efficienti e ridurre i costi di transazione, consentendo di soddisfare la domanda energetica in modo ottimale.

A differenza di altre commodities, sulla Borsa Elettrica non vengono messi sul mercato prodotti o beni di consumo reali contenuti in magazzini, bensì la promessa di produrre in un dato giorno ad una data ora una certa quantità di energia.

La Borsa Elettrica si articola in due segmenti gestiti dal GME, ossia il *Mercato a Pronti (MPE)* e il *Mercato a Termine (MTE)*. Il mercato elettrico comunemente detto è il MPE, che si articola a sua volta in tre sottofasi: *Mercato del Giorno Prima* (*MGP*), l *Mercato Infragiornaliero* (*MI*) e il *Mercato dei Servizi di Dispacciamento* (*MSD*). A complementare il mercato spot, vi sono, inoltre, i *contratti bilaterali*[1] che contribuiscono a migliorare l'efficienza delle transazioni e ad arricchire l'offerta di mercato.[4]

2.2.1 Il Mercato del Giorno Prima

Il Mercato del Giorno Prima (MGP), a cui partecipano produttori, grossisti, l'*Acquirente Unico* (AU) e TERNA , inizia le sue operazioni alle ore 8:00 del nono giorno che precede la data di consegna e termina alle ore 9:00 del ultimo giorno immediatamente antecedente alla consegna. Questo lasso di tempo consente agli operatori di presentare le proprie offerte. Il GME procede, dunque, alla selezione delle offerte basandosi su un sistema d'asta implicito equo, prediligendo quelle più vantaggiose economicamente (priorità di dispacciamento). In particolare, le offerte di vendita sono organizzate per prezzo in ordine ascendente, dalle più economiche

[1] I contratti bilaterali trovano posto nel mercato OTC (Over The Counter) al di fuori del mercato organizzato e standardizzato, tramite i quali soggetti responsabili degli impianti di produzione di energia elettrica possono decidere di cedere l'energia elettrica prodotta e immessa in rete ad un trader/grossista a un prezzo di cessione direttamente negoziato con quest'ultimo.

alle più care e quelle di acquisto in ordine discendente, dalle più vantaggiose a quelle meno convenienti, senza indicazione di prezzo.

L'Acquirente Unico S.p.A., facente parte del gruppo GSE S.p.A., si assume la responsabilità di assicurare l'approvvigionamento di energia elettrica ai clienti che non hanno ancora optato per un fornitore sul mercato libero. Il suo ruolo è quello di acquisire energia sul mercato alle condizioni più favorevoli e di rivenderla ai fornitori di vendita al dettaglio, i quali a loro volta riforniscono utenze domestiche e piccole imprese che non operano nel mercato libero. All'AU sono state inoltre delegate ulteriori funzioni, volte a proteggere i consumatori e a stimolare il processo di liberalizzazione dei mercati dell'energia elettrica e del gas.[5]

Per assicurare il rispetto dei limiti di flusso energetico, TERNA notifica al GME, almeno un'ora prima della conclusione delle contrattazioni nel MGP, i limiti massimi di scambio energetico orario tra diverse aree geografiche e con le zone di interconnessione internazionale. Inoltre, fornisce una previsione della domanda oraria per ogni area geografica e i piani di utilizzo per le unità di produzione secondo la regolamentazione *CIP6/92*[2] [5]. Le offerte che vengono accettate in condizioni di saturazione dei flussi energetici tra aree geografiche risultano in un prezzo di equilibrio stabilito da un algoritmo sviluppato dal GME. Questo algoritmo determina, in presenza di prezzi di vendita differenziati per area, un unico prezzo di acquisto su scala nazionale (PUN), calcolato come media ponderata dei prezzi di vendita per area, basata sui consumi di ciascuna zona. Questo meccanismo considera anche le quantità derivanti dai contratti bilaterali comunicati dagli operatori, che vengono inclusi nel calcolo della quantità di equilibrio.

[2]Il CIP67/92, introdotto dalla delibera del Comitato Interministeriale dei Prezzi il 29 aprile 1992 in risposta alla legge n. 9 del 1991, stabilisce tariffe incentivanti per l'energia prodotta da fonti rinnovabili e fonti considerate "assimilate", ampliando l'ambito di applicazione per includere varie fonti energetiche non specificatamente menzionate nella normativa europea. In virtù di questa delibera, i produttori di energia da fonti rinnovabili o assimilate possono vendere la propria energia al GSE a un prezzo superiore a quello di mercato, posizionandosi quindi in maniera vantaggiosa rispetto alle tradizionali offerte di vendita.

2.2.2 Il Mercato Infra-giornaliero e il Mercato dei Servizi di Dispacciamento

Il Mercato Infra-giornaliero (MI), istituito dalla Legge 02/09 del Ministero dello Sviluppo Economico, ha preso il posto del precedente Mercato di Aggiustamento, offrendo ai produttori, ai grossisti e ai clienti qualificati la possibilità di apportare modifiche ai programmi di iniezione o prelievo di energia stabiliti dal MGP. Attraverso la ripartizione in sette sessioni nell'arco della giornata, il MI facilita lo scambio di energia elettrica destinata al consumo giornaliero, permettendo aggiustamenti tempestivi basati su informazioni aggiornate riguardo la disponibilità delle centrali e il fabbisogno energetico. Le sessioni, che seguono lo schema delle aste implicite simile a quello del MGP, si susseguono a intervalli regolari, consentendo ai partecipanti di ottimizzare la gestione delle centrali di produzione e di aggiornare i programmi di prelievo per rispondere efficacemente alle esigenze del mercato.

Nel caso in cui gli aggiustamenti operati tramite il MI non siano sufficienti o non si riesca a soddisfare il fabbisogno residuo in maniera pronta, entra in gioco il Mercato dei Servizi di Dispacciamento (MSD), che offre una risposta immediata agli squilibri di rete, fornendo energia di regolazione sia positiva che negativa. Il MSD rappresenta il meccanismo attraverso cui TERNA, l'operatore della rete di trasmissione nazionale, acquisisce le risorse necessarie per la gestione e il controllo del sistema elettrico. Gli obiettivi specifici del MSD includono:

- La verifica, per ogni ora e area geografica, della disponibilità di adeguate bande di regolazione per il giorno successivo, apportando se necessario modifiche agli esiti del MGP e del MI per garantirle;
- L'identificazione e risoluzione di eventuali congestioni intra-zonali, modificando gli esiti del MGP per prevenirle;
- Il mantenimento del bilanciamento in tempo reale del sistema elettrico e la definizione dei prezzi per l'energia di bilanciamento, utilizzata in caso di necessità per compensare le variazioni impreviste di offerta e domanda.

Il MSD si struttura in due fasi distinte: una fase anticipatoria (MSD ex-ante), mirata all'acquisto di servizi per risolvere le congestioni identificate dopo il MI e per procurare le risorse necessarie alla regolazione della frequenza; e una fase infra-giornaliera, durante la quale TERNA scambia l'energia necessaria per equilibrare le immissioni e i prelievi in base agli scostamenti dai programmi stabiliti, assicurando così la stabilità del sistema.

2.2.3 Il Mercato a Termine

Il *Mercato a Termine* dell'energia (MTE) rappresenta un pilastro cruciale nell'ecosistema energetico globale, offrendo agli attori del settore un meccanismo fondamentale per la gestione del rischio di prezzo e la pianificazione delle forniture. Attraverso contratti standardizzati per la consegna futura di energia elettrica, questo mercato, di natura speculativa, consente agli operatori di proteggersi dalle fluttuazioni dei prezzi e di stabilire piani di approvvigionamento e produzione più efficienti.

La sua importanza è ulteriormente accentuata nel contesto della transizione verso fonti energetiche più sostenibili, dove la volatilità dei prezzi e la complessità delle forniture richiedono un'attenzione particolare alla gestione del rischio. Attraverso una combinazione di strumenti finanziari e regolamentazione, il mercato a termine dell'energia svolge un ruolo critico nel garantire la stabilità e l'efficienza del sistema energetico, consentendo agli operatori di adattarsi alle mutevoli dinamiche del mercato e di contribuire alla sostenibilità e alla sicurezza dell'approvvigionamento energetico a livello globale.

2.3 Il Prezzo Unico Nazionale (PUN)

Il Prezzo Unico Nazionale dell'energia elettrica rappresenta il fulcro del sistema energetico italiano, fungendo come punto di riferimento per le transazioni all'ingrosso di energia. Determinato quotidianamente dal GME attraverso il MGP, il PUN è il risultato di un equilibrio tra domanda e offerta di energia elettrica.

La sua determinazione si basa su un complesso sistema di aste che riflette non solo le esigenze immediate del mercato ma anche le proiezioni e le strategie degli operatori.
In particolare, questo processo inizia con la raccolta delle offerte di vendita e di acquisto da parte dei partecipanti al mercato, che includono produttori di energia, fornitori, e grandi consumatori, ciascuno dei quali presenta le proprie offerte in termini di quantità di energia e prezzo desiderato o accettabile per le 24 ore del giorno successivo.
Le offerte di vendita sono generalmente formulate dai produttori di energia, che indicano la quantità di energia che intendono vendere e il prezzo minimo a cui sono disposti a cederla. Al contrario, le offerte di acquisto provengono da fornitori e consumatori industriali che specificano la quantità di energia di cui hanno bisogno e il prezzo massimo che sono disposti a pagare.
Terminata la seduta di presentazione delle offerte, il GME attiva il processo per la *risoluzione del mercato*: tutte le offerte di vendita valide ricevute vengono ordinate per prezzo crescente in una curva di offerta aggregata e le offerte di acquisto valide ricevute sono ordinate per prezzo decrescente in una curva di domanda aggregata.

L'intersezione delle due curve determina il prezzo di equilibrio (Figura 2.2) il quale, nel caso in cui i flussi sulla rete derivanti dai programmi non violano nessun limite di transito di energia sugli elettrodotti, è "unico" in tutte le zone.
Tutte le offerte accettate vengono così remunerate al prezzo di equilibrio, e non ai prezzi di vendita minimi e di acquisto massimi indicati in offerta.

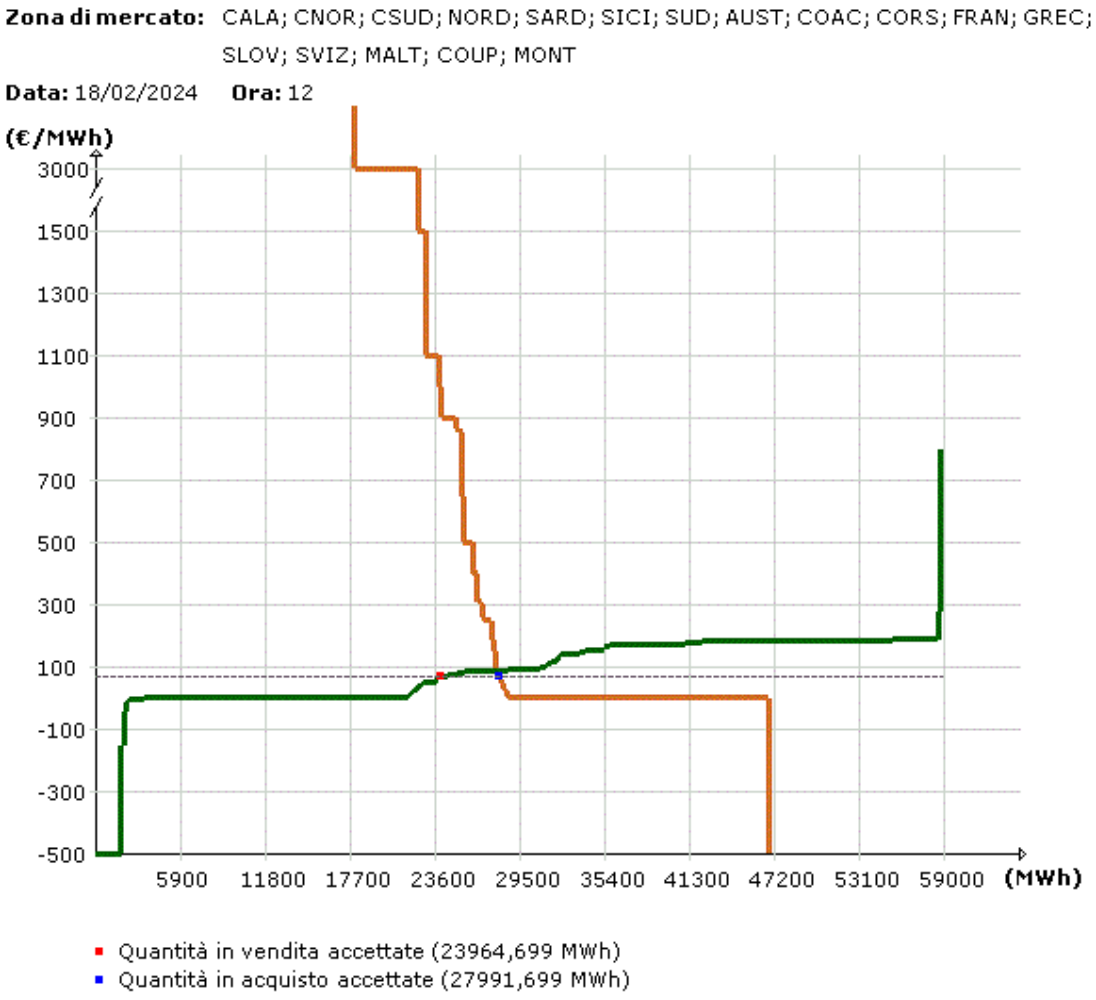

Figura 2.2: Prezzo di equilibrio derivato dall'intersezione delle curve aggregate di acquisot e di vendita (Fonte: GME).

2.3.1 Caratteristiche della serie temporale del PUN

A differenza di altre commodities, l'energia elettrica presenta la peculiarità di non essere stoccabile facilmente con le tecnologie attuali, rendendo la sua gestione unica.
Questa caratteristica impone una sincronizzazione quasi perfetta tra produzione e consumo, con le centrali elettriche che devono regolare la loro attività in tempo reale per soddisfare le fluttuazioni della domanda.

Un altra caratteristica che contraddistingue il PUN, è la sua volatilità significativa, guidata da una varietà di fattori tra cui la periodicità della domanda, le condizioni meteorologiche e la disponibilità di fonti di generazione.
La periodicità si manifesta su scala giornaliera, con picchi durante le ore di punta e su scala stagionale, con variazioni legate al clima e alle abitudini di consumo (Figura 2.3). Infine, eventi imprevisti come guasti agli impianti o cali improvvisi nella produzione da fonti rinnovabili possono causare "spikes" di prezzo, evidenziando la sensibilità del sistema a fluttuazioni immediate della domanda e dell'offerta.

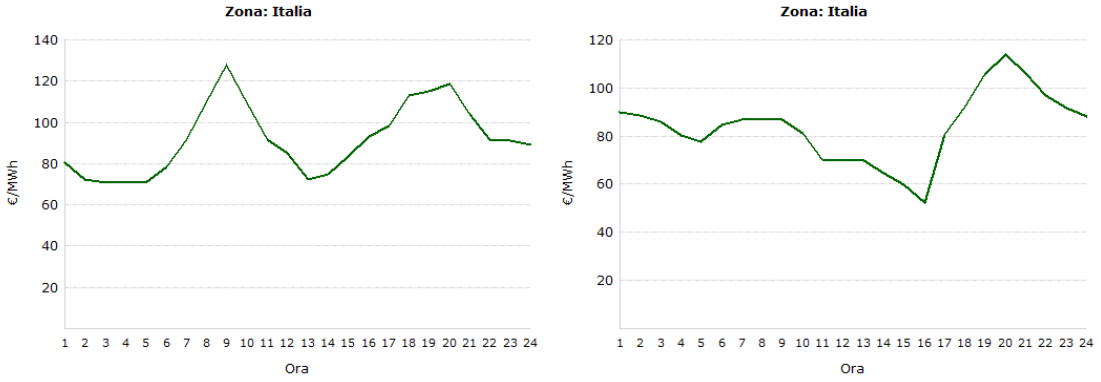

Figura 2.3: Tipico andamento del PUN in Italia nei giorni feriali (a sinistra) e nei giorni festivi (a destra) (Fonte: GME).

La volatilità del PUN ha importanti ripercussioni sui consumatori influenzando i prezzi al dettaglio dell'energia elettrica e, di conseguenza, le bollette energetiche. Per i consumatori nel mercato libero, comprendere le dinamiche del PUN può offrire, dunque, opportunità di risparmio, ad esempio attraverso la scelta di tariffe basate su prezzi variabili o l'adozione di soluzioni di autoconsumo e accumulo.
Parallelamente, occorre che le politiche energetiche e le regolamentazioni tengano conto della volatilità del PUN al fine di promuovere la sostenibilità, l'efficienza e la sicurezza dell'approvvigionamento, stimolando investimenti in infrastrutture, innovazione tecnologica e integrazione di fonti rinnovabili. L'integrazione crescente di

quest'ultime nel mix energetico italiano, infatti, avrebbe evidenti effetti significativi sul valore del PUN, in termini assoluti. Tuttavia, sebbene la generazione da fonti solari e eoliche, caratterizzata da costi marginali bassi o nulli, possa contribuire a ridurre il prezzo dell'energia nelle ore di maggiore produzione, introdurrebbe, data la naturale variabilità di quest'ultime, nuove sfide nella gestione della rete e nella previsione dei prezzi.

In conclusione, emerge chiaramente l'importanza di un approccio olistico che consideri molteplici variabili in gioco. Dalle politiche energetiche alle innovazioni tecnologiche, dall'impatto delle condizioni meteorologiche alle strategie di mercato, ogni fattore contribuisce a modellare il panorama in cui il PUN si determina.
Per questa ragione, condurre analisi di previsione di tale grandezza può risultare difficoltoso e complesso richiedendo l'adozione di tecnologie sempre più avanzate di analisi dati e Machine Learning [6] [7].

Capitolo 3

Metodologia

Questa sezione offre una breve presentazione della letteratura esistente a supporto dei modelli previsionali e una panoramica dettagliata sui dati disponibili presentando la strategia metodologica adottata per la previsione del Prezzo Unico Nazionale.

In particolare, nei capitoli successivi, verranno esplorati i seguenti modelli:

1. **SARIMA (Seasonal Autoregressive Integrated Moving Average)**;
2. **Multiple linear regression**;
3. **XGBoost (Extreme Gradient Boosting)**.

Verrà, dapprima, presentato il linguaggio di programmazione utilizzato con le relative librerie, a cui seguirà una fase iniziale di reperimento ed elaborazione delle variabili impiegate nelle analisi.
Successivamente, verranno esplorate le architetture matematiche dei modelli impiegati e illustrato il processo di addestramento e validazione.
Verrà, infine, condotta una valutazione delle metriche di performance al fine di individuare il modello che verrà impiegato nell'analisi previsionale.

3.1 Stato dell'Arte: Tecniche di Previsione del Prezzo Unico Nazionale

La capacità di prevedere accuratamente il PUN assume un'importanza vitale sia per i produttori, che ottimizzano le operazioni degli impianti per incrementare i loro guadagni, sia per i consumatori, interessati a ridurre le spese energetiche.
Esistono numerosi metodi per la previsione del Prezzo Unico Nazionale, ognuno rientrante in due specifiche categorie:[8]

1. **Modelli Statistici**;
2. **Modelli Machine Learning**

3.1.1 Modelli Statistici

Tra i modelli previsionali statistici più efficaci per prevedere i prezzi dell'energia vi sono quelli impiegati per analizzare le serie temporali, includendo tecniche come la *Regressione Lineare*, l'*AutoRegressive* (AR), la *Moving Average* (*MA*), l'*AutoRegressive Moving Average* (*ARMA*), l'*AutoRegressive Integrated Moving Average* (*ARIMA*), il modello *Seasonal AutoRegressive Integrated Moving Average* (*SARIMA*) e il modello *Generalized AutoRegressive Conditional Heteroschedasticity* (*GARCH*).

La *Regressione Lineare* è ampiamente adottata per la previsione dei prezzi dell'energia elettrica e si pone l'obiettivo di identificare le relazioni tra una variabile dipendente e un insieme di variabili indipendenti. Nel Capitolo 5, verrà approfondita l'architettura matematica della Regressione Lineare.

Il modello *AR* prevede il prezzo corrente basandosi su una combinazione lineare dei valori storici aggiungendo un termine di errore stocastico, mentre il modello *MA* lo modella come una somma del valore medio e di una media mobile degli errori stocastici.

L'*ARIMA* estende l'*ARMA* per includere i processi di differenziazione al fine di rendere la serie stazionaria, utilizzando sia i valori storici che gli errori passati per prevedere il futuro. Il modello SARIMA, infine, aggiunge un termine che tiene conto della "stagionalità" della serie analizzata. Quest'ultimo verrà approfondito nel Capitolo 4.

Laddove la varianza e covarianza dei dati risulti non costante nel tempo e , pertanto, i residui di un eventuale modello statistico sulla media risultino eteroschedastici, i modelli GARCH potrebbero contribuire a prevedere la variabilità dei prezzi.

Sebbene i modelli statistici siano ampiamente applicati e riconosciuti per la loro efficacia in numerosi campi, presentano delle limitazioni nella cattura di pattern non lineari delle serie storiche. Essi, infatti, tendono a essere accurati solo quando le variazioni dei prezzi seguono le tendenze già esplorate nei dati di addestramento, ignorando l'effetto di fattori esterni come dinamiche di mercato o variazioni ambientali. Inoltre, funzionano meglio con dati a bassa frequenza, potendo invece incontrare difficoltà nel fronteggiare rapidi cambiamenti di prezzo.

Altri modelli statistici che sono stati impiegati per la previsione di prezzi di mercato sono:

- Il modello *Jump-Diffusion*: integra salti improvvisi nei prezzi o nelle rendite, oltre alle variazioni continue tipiche dei modelli di diffusione. Questo lo rende particolarmente adatto a catturare eventi di mercato improvvisi e rari (come crisi finanziarie o annunci economici inaspettati) che possono influenzare significativamente i prezzi dell'energia.

- Il modello di *Regressione Quantile*: a differenza della regressione lineare che stima il valore medio di una variabile dipendente basata su variabili indipendenti, la regressione quantile mira a prevedere un determinato quantile della distribuzione della variabile dipendente. Questo approccio è utile per analizzare la relazione tra variabili quando l'interesse è focalizzato su diversi punti della distribuzione, come i valori estremi (es. i prezzi dell'energia durante picchi di domanda).

- Il modello *GAM* (*General Additive Model*): flessibile estensione dei modelli lineari generalizzati, permettono di modellare relazioni non lineari tra la variabile target e una o più variabili predittive. Attraverso funzioni di "smoothing", i GAM possono adattarsi a complessi schemi di dati, rendendoli adatti per analizzare serie temporali di prezzi dell'energia con pattern stagionali o ciclici.

- Il metodo *ESM* (*Exponential Smoothing Method*): tecnica di previsione che pondera in modo esponenziale i dati più recenti, attribuendo meno importanza ai dati più vecchi. Questo metodo è particolarmente efficace per serie temporali con tendenze e stagionalità, offrendo previsioni su breve termine per i prezzi dell'energia che si adattano rapidamente ai cambiamenti del mercato.

3.1.2 Modelli Machine Learning

Nel campo dell'intelligenza computazionale, i modelli *Machine Learning* si distinguono principalmente in due famiglie: i modelli *Supervisionati* e i modelli *Non Supervisionati*.

I modelli supervisionati imparano da dati già etichettati, puntando a prevederne o classificarne di nuovi basandosi su esempi passati. Fra questi rientrano algoritmi come le *Reti Neurali Artificiali* (*ANN*), *Alberi Decisionali*, modelli *Regressivi* o di *Classificazione* i quali tentano di identificare le relazioni tra variabili dipendenti e indipendenti.
I modelli non supervisionati, invece, esplorano dati non etichettati per identificarne strutture o raggruppamenti nascosti. Fra questi è possibile identificare analisi di *Clustering* e *Regole di Associazione*

Modelli Ensemble Learning

Nell'ambito dei modelli di Machine Learning, esiste una categoria nota come *Modelli di Insieme* o *Ensemble Learning*.

I modelli Ensemble Learning rappresentano un'avanzata metodologia nell'ambito della previsione e dell'analisi predittiva, particolarmente efficaci nella risoluzione di problemi complessi.
L'approccio Ensemble si basa sull'idea di combinare più modelli predittivi al fine di migliorare l'accuratezza e la robustezza delle previsioni rispetto a quanto sarebbe stato possibile ottenere con un singolo modello; sfrutta, dunque, i punti di forza di vari algoritmi riducendo al contempo l'effetto delle loro singole debolezze.

Combinando le previsioni di più modelli, i metodi Ensemble tendono a ridurre tre tipi principali di errore (*bias*, *varianza*, e *rumore*) offrendo così previsioni più accurate. Essendo, inoltre, meno sensibili alle fluttuazioni dei dati di addestramento riducono il rischio di *overfitting*.

Esistono diverse tipologie di Ensemble, ognuna con le proprie caratteristiche e modalità di implementazione. Le principali categorie includono:

1. **Bagging**: aumenta la stabilità e riduce la varianza costruendo più modelli indipendenti su diversi sottoinsiemi di dati, ottenuti tramite campionamento con sostituzione (ossia che i singoli punti di dati possono essere scelti più di una volta) dal set di addestramento originale (*Bootstrap*). Le previsioni finali

sono ottenute aggregando le previsioni di tutti i modelli (mediante votazione per classificazione o per regressione). Il *Random Forest*, che costruisce diversi alberi decisionali su vari sottoinsiemi di dati e caratteristiche, è uno di questi;

2. **Boosting**: costruisce modelli in modo sequenziale, in cui ogni modello successivo cerca di correggere gli errori del modello precedente. L'obiettivo è creare un modello robusto e accurato combinando più modelli deboli. A tal proposito si citano i modelli *AdaBoost* (*Adaptive Boosting*) e *Gradient Boosting* incluso il suo derivato *XGBoost*, che verrà approfondito nel Capitolo 6;

3. **Stacking** (**Stacked Generalization**): diversi modelli di base vengono addestrati sui dati originali, e un modello "meta" (o un modello di livello superiore) viene addestrato per aggregare le previsioni dei modelli di base. La peculiarità dello "Stacking" è l'uso di un modello di apprendimento per combinare le previsioni. Infatti, a differenza del modello Bagging in cui l'output era il risultato di una votazione, nello Stacking viene introdotto un ulteriore classificatore (dmeta-classificatore) che utilizza le predizioni di altri sotto-modelli per effettuare un ulteriore apprendimento.

3.2 Linguaggio di Programmazione

Il linguaggio di programmazione impiegato per l'implementazione dei modelli è `Python`, noto per la sua estrema versatilità e per l'ampio ecosistema di librerie dedicate all'analisi dei dati, alla statistica e all'apprendimento automatico.

Tra le principali librerie utilizzate nel progetto, si annoverano:

- **Matplotlib e Seaborn**, le quali hanno fornito strumenti indispensabili per la realizzazione di grafici e visualizzazioni dati. `Matplotlib` si è distinto per le sue capacità di plotting estremamente versatile, mentre `Seaborn` ha facilitato la generazione di grafici statistici di elevata qualità visiva, basandosi sulle funzionalità di Matplotlib;
- **Pandas**, che ha permesso una manipolazione e analisi dei dati efficiente e versatile, facilitando operazioni quali la lettura, la scrittura e la trasformazione di dati strutturati;
- **Numpy**, impiegata per il suo supporto avanzato ai calcoli matematici, in particolare per la gestione ottimizzata di array e matrici, che ha garantito l'esecuzione di operazioni numeriche con elevata efficienza;
- **Statsmodels**, che ha giocato un ruolo cruciale nell'implementazione di modelli statistici avanzati, inclusi quelli per l'analisi di regressione e la modellazione delle serie storiche, essenziali per le previsioni del Prezzo Unico Nazionale attraverso l'uso di modelli ARIMA e SARIMAX;
- **Scikit-learn**, leader nell'apprendimento automatico, è stata utilizzata per diverse funzionalità, tra cui la regressione lineare, la divisione dei dati in set di addestramento e di test e il calcolo di metriche di valutazione delle performance dei modelli, come il MAE e il RMSE;
- **XGBoost**, selezionata per l'implementazione di tecniche di boosting;
- **Plotly**, utilizzata, infine, per arricchire il progetto con grafici interattivi, migliorando l'accessibilità e la comprensione dei risultati ottenuti.

3.3 Preprocessing dei Dati

La fase preliminare di ogni analisi predittiva comporta un'attenta preparazione dei dati, essenziale per garantirne la consistenza e l'accuratezza. Esso rappresenta uno step fondamentale poiché la performance del modello dipende strettamente dalla bontà dei dati con cui esso viene addestrato ("*garbage-in garbage-out*").

Questo processo inizia con l'identificazione e la successiva gestione dei valori mancanti, imputati attraverso l'interpolazione lineare, per assicurare che il data-set sia completo e pronto per l'analisi.

Successivamente è stata eseguita un'identificazione grafica degli outliers tramite BoxPlot il quale rappresenta un ottimo strumento grafico per l'identificazione dei valori anomali generalmente visualizzati come punti che cadono al di fuori dei "baffi" del grafico. L'identificazione è eseguita mediante il metodo interquartile (IQR). L'IQR rappresenta la differenza tra il 75° percentile (Q3) e il 25° percentile (Q1) dei dati. Gli outlier, dunque, possono essere identificati come i valori che cadono al di sotto di $Q1 - 1.5IQR$ o al di sopra di $Q3 + 1.5IQR$. Tuttavia, occorre precisare che i valori anomali non sono stati eliminati al fine di evitare di rimuovere dati che riflettessero variazioni naturali delle variabili coinvolte.

Il valore del prezzo dell'energia, come visto in precedenza, è influenzato da numerosi fattori, ma solo alcuni di questi sono stati considerati nello sviluppo dei modelli qui descritti.
I dati che sono stati reperiti sono quelli relativi al PUN (la variabile endogena, ossia dipendente) e variabili esogene (denominate Features) ossia indipendenti come: *fabbisogno elettrico* (*DEM*), *energia generata da eolico* (*WIND*), *energia generata da fotovoltaico* (*SOLAR*) e *prezzo MGP del Gas naturale* (*GAS*).
In particolare, il modello SARIMA avrà come unica variabile (endogena) quella del PUN, mentre nei modelli Multiple Linear Regression e XGBoost, verranno inserite nel data-set anche le variabili dipendenti DEM, WIND, SOLAR e GAS.

Prima di procedere con l'analisi, occorre fare una doverosa promessa sul periodo temporale scelto.
In particolare, i dati collezionati riguardano il periodo temporale che va dal 2015 al 2021 per le seguenti motivazioni:

1. I dati relativi al prezzo del GAS presentano una disomogeneità temporale per il periodo antecedente il 2015. Ciò ha portato a impostare come limite inferiore del data-set il **01/01/2015 00:00:00**;

2. Il contesto geopolitico e sanitario che la popolazione italiana si è ritrovata ad affrontare nell'ultimo triennio ha portato ad un aumento vertiginoso e anomalo dei prezzi del gas naturale e, di conseguenza, dell'energia elettrica. Dal momento che tali aumenti si discostano notevolmente dai valori medi della serie si è ritenuto opportuno considerare come limite superiore del data-set il **31/03/2021 23:00:00**.

3.3.1 PUN

Il GME [9] fornisce dati dettagliati riguardanti il PUN per diverse zone geografiche, coprendo periodi dal 2004 al 2024. La preparazione di questi dati ha richiesto l'unificazione delle informazioni in un unico data-frame, assicurando una continuità temporale.

Utilizzando la libreria *Pandas*, il processo è iniziato con il caricamento dei file *".xlsx"* selezionando specificatamente il foglio *"Prezzi-Prices"* al fine di considerare solo le colonne relative alla data, all'ora e ai prezzi. Successivamente, i fogli Excel di ciascun anno sono stati uniti in un unico data-frame, garantendo così la continuità temporale dei dati dal 2015 al 2021.

A tal proposito, le colonne della data e dell'ora sono state unificate in un'unica colonna denominata `Data e Ora`, formattata nel formato `DD-MM-YYYY hh:mm:ss`, successivamente indicizzata mediante il comando `datetime.index` al fine di trasformare la serie in una struttura adatta alle analisi temporali.

Infine, i data-frame finali sono stati esportati e salvati in un file Excel denominato "PUN_15_21.xlsx", pronto per future analisi.

Tabella 3.1: PUN

Data e Ora	PUN
2015-01-01 00:00:00	52.327563
2015-01-01 01:00:00	49.892778
2015-01-01 02:00:00	39.100000
2015-01-01 03:00:00	35.870000
2015-01-01 04:00:00	33.400000
2021-03-31 19:00:00	55.12492
2021-03-31 20:00:00	60.90000
2021-03-31 21:00:00	67.54000
2021-03-31 22:00:00	91.93131
2021-03-31 23:00:00	81.59987

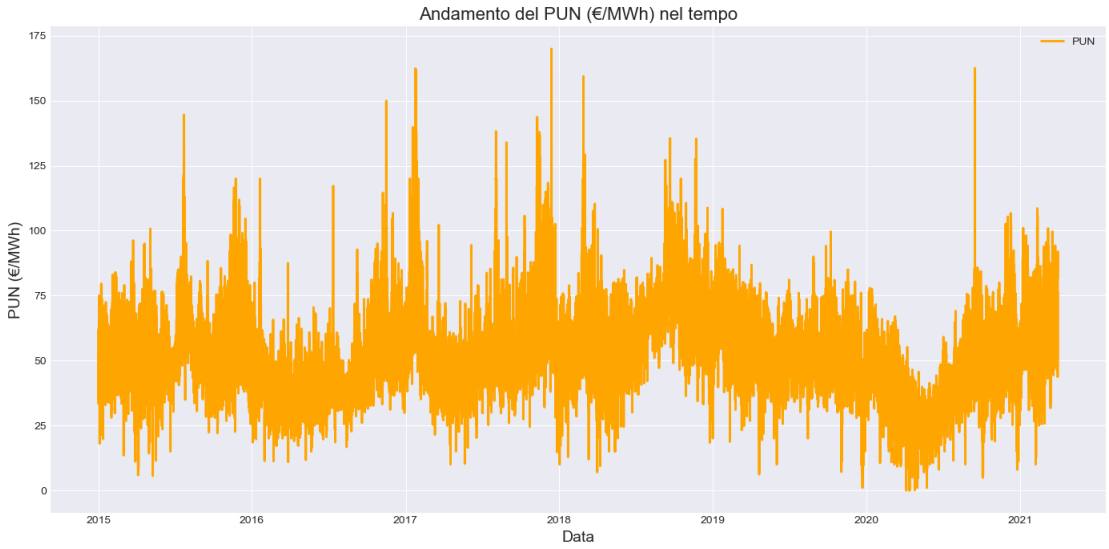

Figura 3.1: Andamento del PUN dal 2015 al 2021

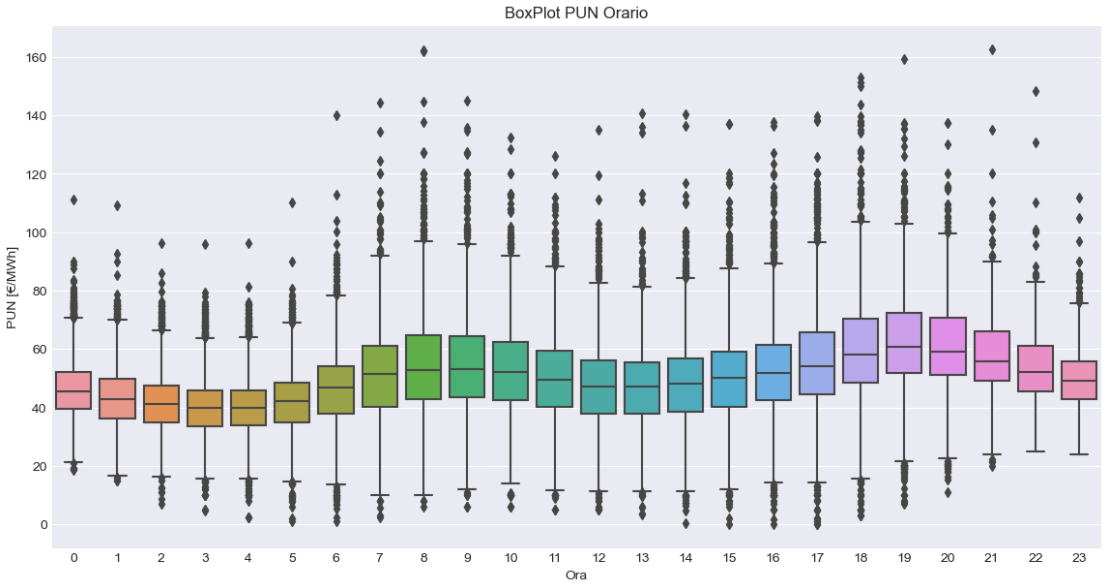

Figura 3.2: Outlier PUN - orario

3.3.2 DEM, SOLAR e WIND

I dati DEM, SOLAR E WIND sono forniti dal database *ENTSO-E* [10], una piattaforma trasparente che mira a fornire accesso gratuito e continuo ai dati del mercato elettrico europeo per tutti gli utenti, attraverso sei categorie principali: fabbisogno, generazione, trasmissione, bilanciamento, interruzioni e gestione della congestione.
I dati relativi alla domanda di energia sono forniti in intervalli quarto-orari; perciò è stato ritenuto necessario convertirli in dati orari mediante il calcolo della media oraria, al fine di poterli successivamente integrare nel data-set originale contenente il PUN.

Tabella 3.2: DEM

Data e Ora	DEM
2015-01-01 00:00:00	24405.00
2015-01-01 01:00:00	23126.00
2015-01-01 02:00:00	21534.00
2015-01-01 03:00:00	20219.00
2015-01-01 04:00:00	19470.00
2021-03-31 19:00:00	37914.00
2021-03-31 20:00:00	41071.00
2021-03-31 21:00:00	42255.74
2021-03-31 22:00:00	38741.99
2021-03-31 23:00:00	34969.74

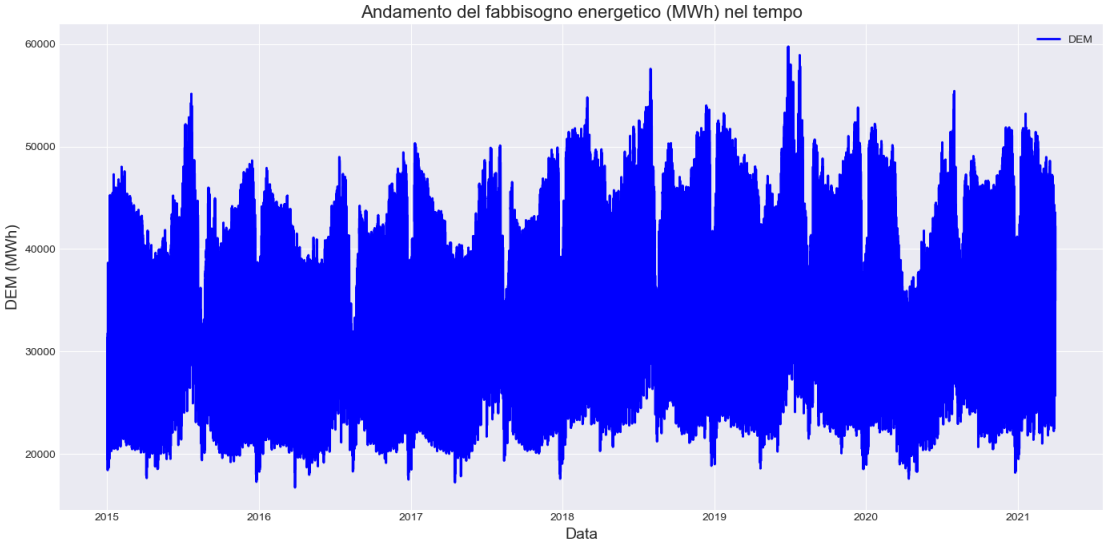

Figura 3.3: Andamento del fabbisogno elettrico (DEM) dal 2015 al 2021

Tabella 3.3: SOLAR E WIND

Data e Ora	SOLAR	WIND
2015-01-01 00:00:00	0.0	5367.0
2015-01-01 01:00:00	0.0	3660.0
2015-01-01 02:00:00	0.0	3670.0
2015-01-01 03:00:00	0.0	3714.0
2015-01-01 04:00:00	0.0	4133.0
2021-03-31 19:00:00	0.0	2412.0
2021-03-31 20:00:00	0.0	2550.0
2021-03-31 21:00:00	0.0	2602.0
2021-03-31 22:00:00	0.0	2622.0
2021-03-31 23:00:00	0.0	2244.0

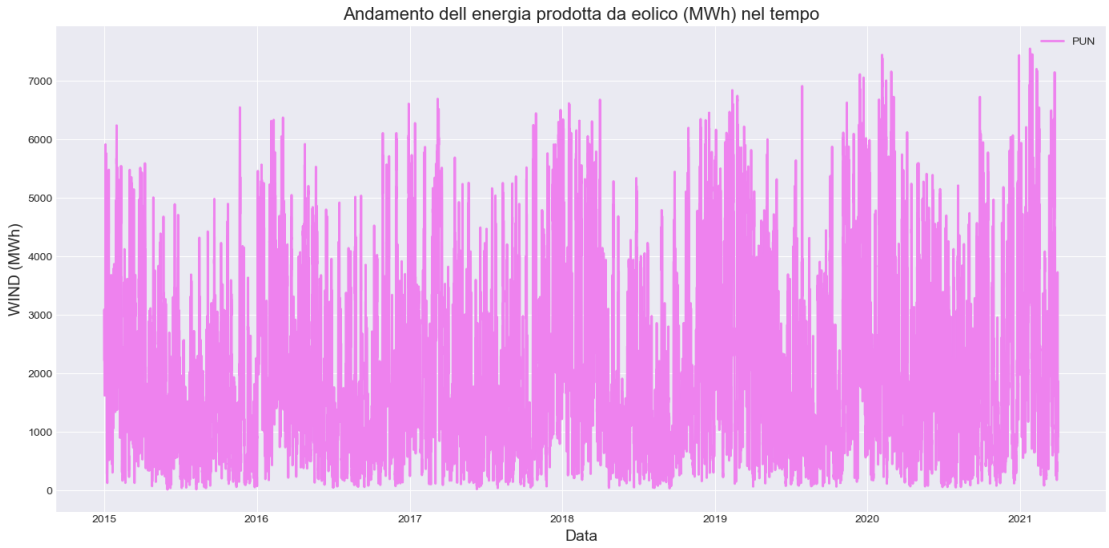

Figura 3.4: Andamento dell'energia prodotta da eolico (WIND) dal 2015 al 2021

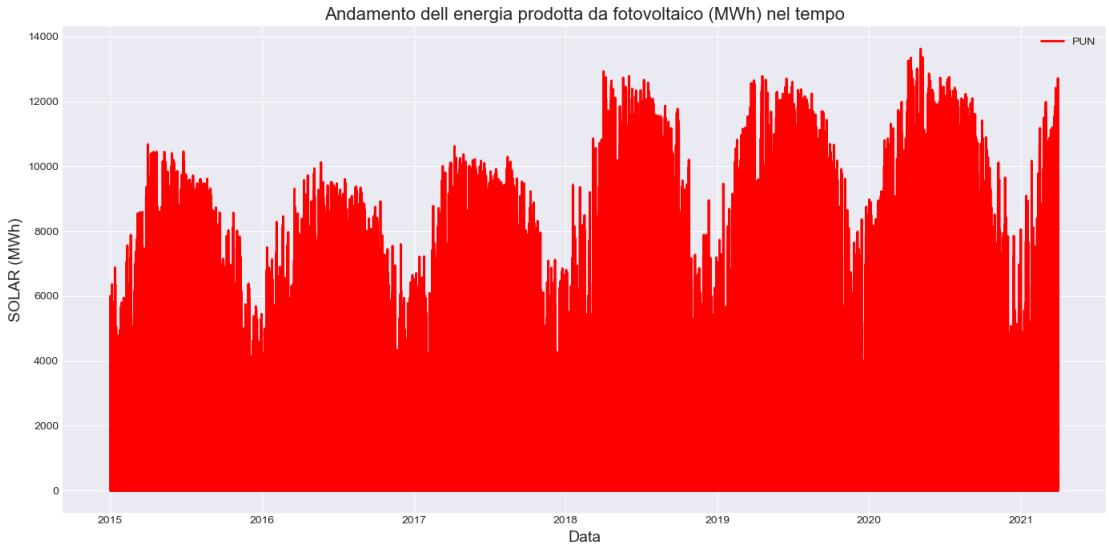

Figura 3.5: Andamento dell'energia prodotta da fotovoltaico (SOLAR) dal 2015 al 2021

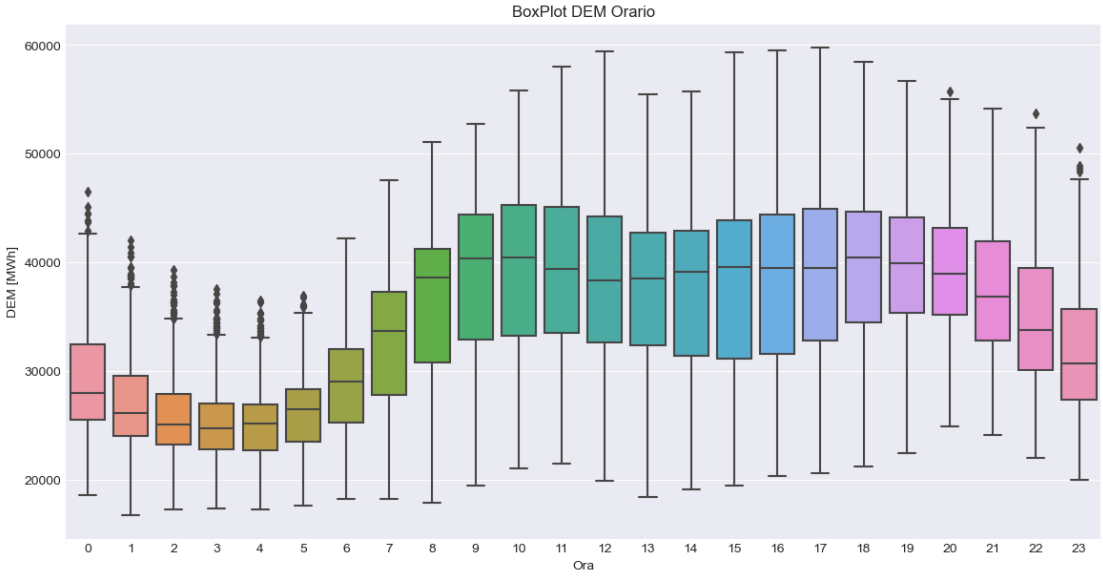

Figura 3.6: Outlier DEM - orario

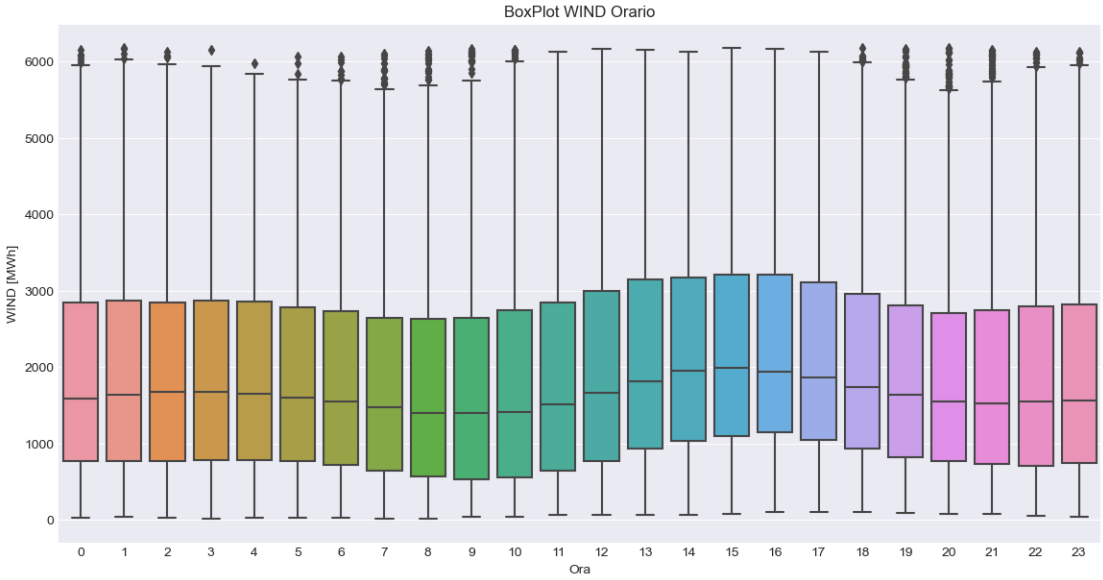

Figura 3.7: Outlier WIND - orario

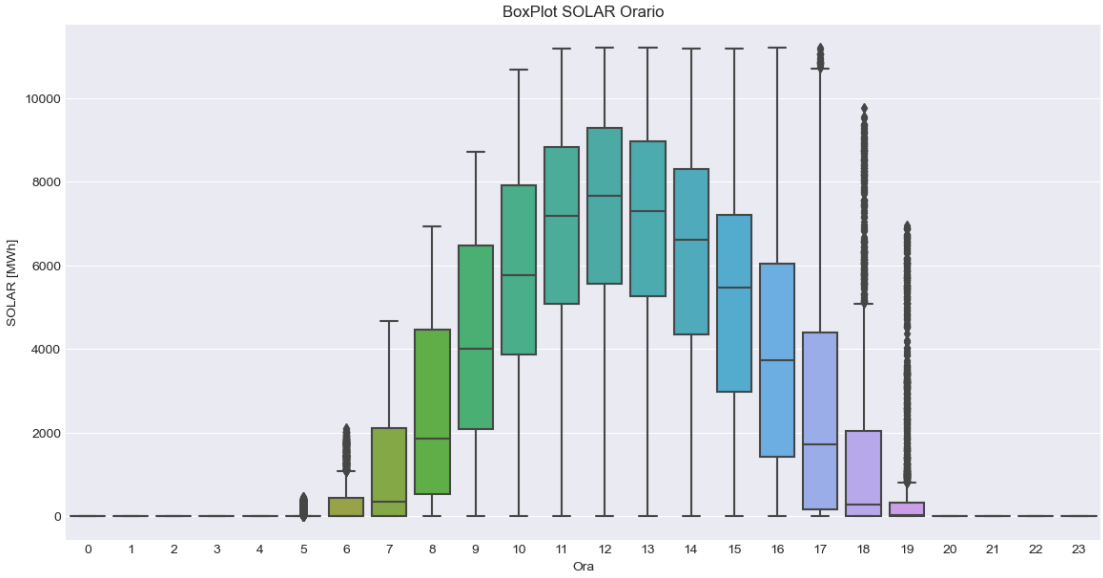

Figura 3.8: Outlier SOLAR - orario

3.3.3 GAS

In merito al costo del gas, il GME [9] fornisce dettagliati dati relativi a ciascuna zona geografica e virtuale, compresi i volumi di gas venduti, non venduti e acquistati. Questi valori sono registrati per gli anni termici dal 2004 al 2023. I dati originariamente forniti erano in formato giornaliero, quindi è stato necessario trasformarli in dati orari, distribuendo il valore costante giornaliero in ogni ora del giorno.

Per affrontare il problema dei dati mancanti, è stata applicata un'interpolazione lineare consentendo di mantenere la coerenza temporale dei dati. A seguito dell'interpolazione, è stata verificata l'assenza di ulteriori valori mancanti nella colonna GAS, garantendo così l'integrità del data-set per le successive analisi.

Tabella 3.4: GAS

Data e Ora	GAS
2015-01-01 00:00:00	20.107
2015-01-01 01:00:00	20.107
2015-01-01 02:00:00	20.107
2015-01-01 03:00:00	20.107
2015-01-01 04:00:00	20.107
2021-03-31 19:00:00	36.945
2021-03-31 20:00:00	36.945
2021-03-31 21:00:00	36.945
2021-03-31 22:00:00	36.945
2021-03-31 23:00:00	36.945

In conclusione, i dati relativi ad ogni anno dal 2015 al 2021 sono stati raggruppati in un unico data-set (Tabella 3.5).

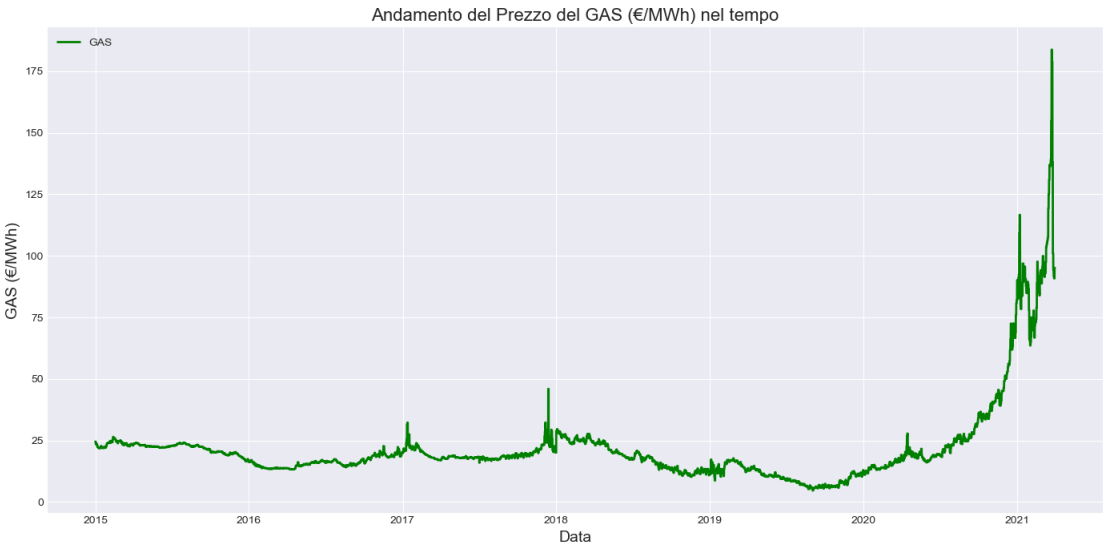

Figura 3.9: Andamento del prezzo del GAS naturale dal 2015 al 2021

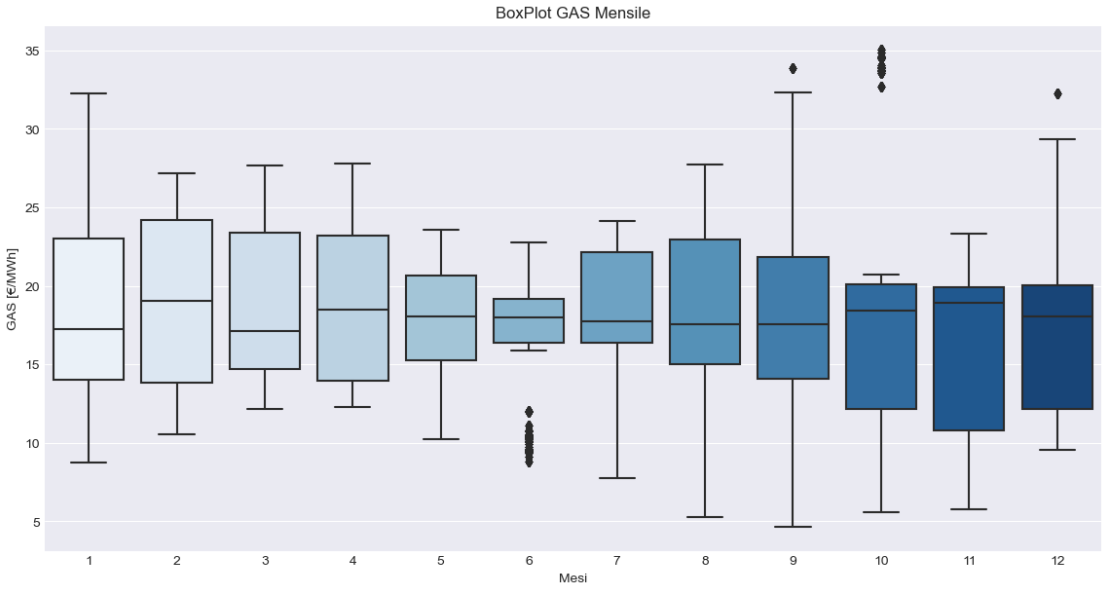

Figura 3.10: Outlier GAS - mensile

Tabella 3.5: Data-set completo

Data e Ora	PUN	DEM	SOLAR	WIND	GAS
2015-01-01 00:00:00	52.327563	24405.00	0.0	5367.0	20.107
2015-01-01 01:00:00	49.892778	23126.00	0.0	3660.0	20.107
2015-01-01 02:00:00	39.100000	21534.00	0.0	3670.0	20.107
2015-01-01 03:00:00	35.870000	20219.00	0.0	3714.0	20.107
2015-01-01 04:00:00	33.400000	19470.00	0.0	4133.0	20.107
2021-03-31 19:00:00	55.12492	37914.00	0.0	2412.0	36.945
2021-03-31 20:00:00	60.90000	41071.00	0.0	2550.0	36.945
2021-03-31 21:00:00	67.54000	42255.74	0.0	2602.0	36.945
2021-03-31 22:00:00	91.93131	38741.99	0.0	2622.0	36.945
2021-03-31 23:00:00	81.59987	34969.74	0.0	2244.0	36.945

3.4 Suddivisione in Training e Testing Set

Per la validazione del modello predittivo, i dati sono stati suddivisi in un set di addestramento (training set), che include i dati fino al 31 Marzo 2020, e un set di valutazione (test set), che comprende i dati successivi a questa data fino al 31 Marzo 2021.

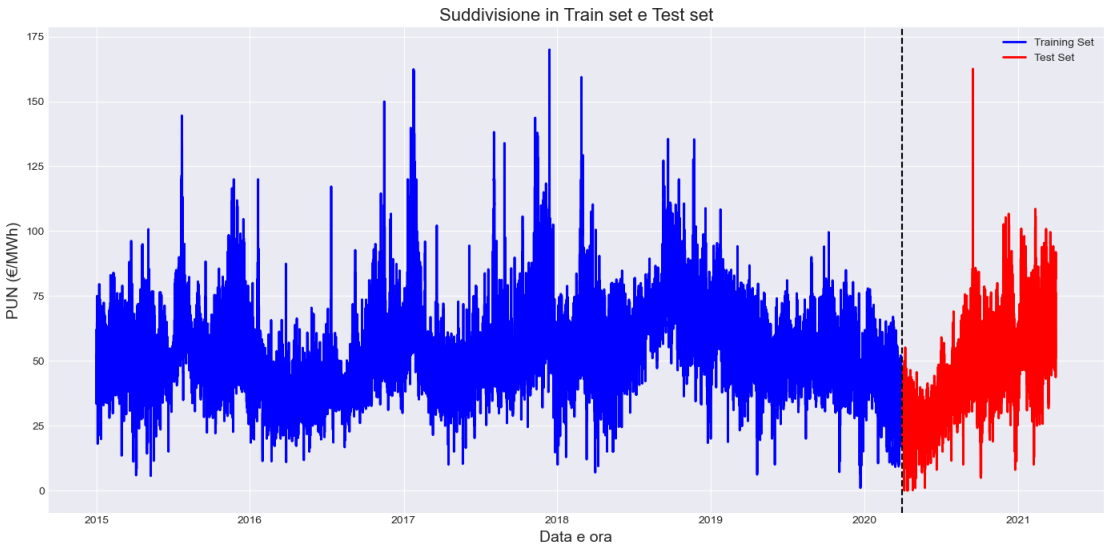

Figura 3.11: Divisione dei dati in training set e test set

Capitolo 4

Seasonal Auto-Regressive Integrated Moving Average (SARIMA)

Il modello SARIMA (Seasonal AutoRegressive Integrated Moving Aerage) rappresenta un'estensione del modello ARIMA (AutoRegressive Integrated Moving Average), progettato specificamente per catturare la stagionalità nei dati delle serie temporali. [11] [12]

Il modello è identificato da sette parametri: $(p, d, q) \times (P, D, Q)_s$, dove:

- p indica il numero di termini autoregressivi;
- d rappresenta il grado di differenziazione;
- q denota il numero di termini della media mobile;
- P, D, e Q corrispondono rispettivamente ai parametri autoregressivi stagionali, al grado di differenziazione stagionale e ai termini della media mobile stagionale;
- s indica il periodo di stagionalità.

4.1 Architettura Matematica

Il modello SARIMA combina sia le componenti non stagionali (ARIMA) sia le componenti stagionali per modellare serie temporali complesse. La sua architettura può essere distinta in tre parti principali:

1. **Componente Autoregressiva (AR)**: prevede i valori futuri in base a una combinazione lineare dei valori passati della serie;
2. **Componente di Differenziazione (I)**: viene utilizzata per rendere la serie temporale stazionaria, ovvero per rimuovere tendenze e ciclicità, facilitando così la modellazione della serie;
3. **Componente Media Mobile (MA)**: componente che utilizza la relazione tra un valore osservato e un insieme di errori casuali provenienti da valori osservati precedenti per modellare la serie temporale.

Le componenti stagionali P, D, e Q introducono termini aggiuntivi per catturare le fluttuazioni stagionali.

La formulazione matematica integra i comportamenti sia stagionali sia non stagionali della serie temporale. La serie differenziata d volte e D volte a livello stagionale può essere espressa come:

$$\Phi_P(B^s)\phi_p(B)\nabla^d\nabla_s^D y_t = \theta_q(B)\Theta_Q(B^s)\epsilon_t$$

dove:

- B è l'operatore di backshift (ritardo),
- ∇ è l'operatore di differenziazione non stagionale,
- ∇_s è l'operatore di differenziazione stagionale,
- ϕ e Φ sono i polinomi autoregressivi non stagionali e stagionali,
- θ e Θ sono i polinomi della media mobile non stagionali e stagionali,
- y_t rappresenta il valore della serie temporale al tempo t,
- ϵ_t è il termine di errore al tempo t.

4.2 Processo di Calcolo

4.2.1 Test della Stazionarietà

Tramite il pacchetto `adfuller` della libreria `statsmodels.tsa.stattools` è stato eseguito il test di *Dickey-Fuller Aumentato* (ADF) per verificare la stazionarietà delle serie temporali.
Questo test è fondamentale per determinare se una serie temporale presenta una radice unitaria.
Il concetto di radice unitaria in una serie temporale si riferisce a una situazione in cui la serie non è stazionaria. Se una serie temporale ha una radice unitaria, ciò significa che presenta un pattern o "trend" che persiste e che possiede varianza e media non costanti nel tempo. Questo comportamento risulta problematico per molte analisi statistiche dal momento che i metodi classici presuppongono la stazionarietà dei dati.

Come si evince dai risultati riportati in Tabella 4.1, il test ha rifiutato l'ipotesi nulla di radice unitaria e di conseguenza la serie temporale del PUN risulta essere stazionaria e adatta al modello SARIMA.
Se il test avesse confermato la non stazionarietà della serie, sarebbe stato necessario differenziare quest'ultima per renderla stazionaria. Tramite differenziazione, infatti, la serie viene trasformata sottraendo il valore corrente con quello precedente. Questo può essere fatto più volte fino a quando la serie non diventa stazionaria.

Tabella 4.1: Test ADF per la stazionarietà - PUN

Augmented Dickey-Fuller Test	
ADF test statistic	$-1{,}08\text{E}+07$
p-value	$1{,}58\text{E}-13$
lags used	$5{,}90\text{E}+07$
observations	$5{,}47\text{E}+10$
critical value (1\%)	$-3{,}43\text{E}+06$
critical value (5\%)	$-2{,}86\text{E}+06$
critical value (10\%)	$-2{,}57\text{E}+06$
Strong evidence against the null hypothesis	
Reject the null hypothesis	
Data has no unit root and is stationary	

4.2.2 Funzioni di Autocorrelazione (ACF) e Autocorrelazione Parziale(PACF)

L'autocorrelazione e l'autocorrelazione parziale sono misure dell'associazione tra i valori della serie corrente e quelli delle serie passate che indicano quali sono i valori delle serie passate più utili per prevedere valori futuri.
Conoscendo questo dato è possibile determinare l'ordine dei processi nel modello SARIMA. Più precisamente:

- **Funzione di Autocorrelazione (ACF)**: al ritardo k, essa è la correlazione tra i valori della serie che sono separati da k intervalli.
- **Funzione di Autocorrelazione Parziale (PACF)**: al ritardo k, essa è la correlazione tra i valori della serie che sono separati da k intervalli, tenuto conto dei valori degli intervalli intermedi.

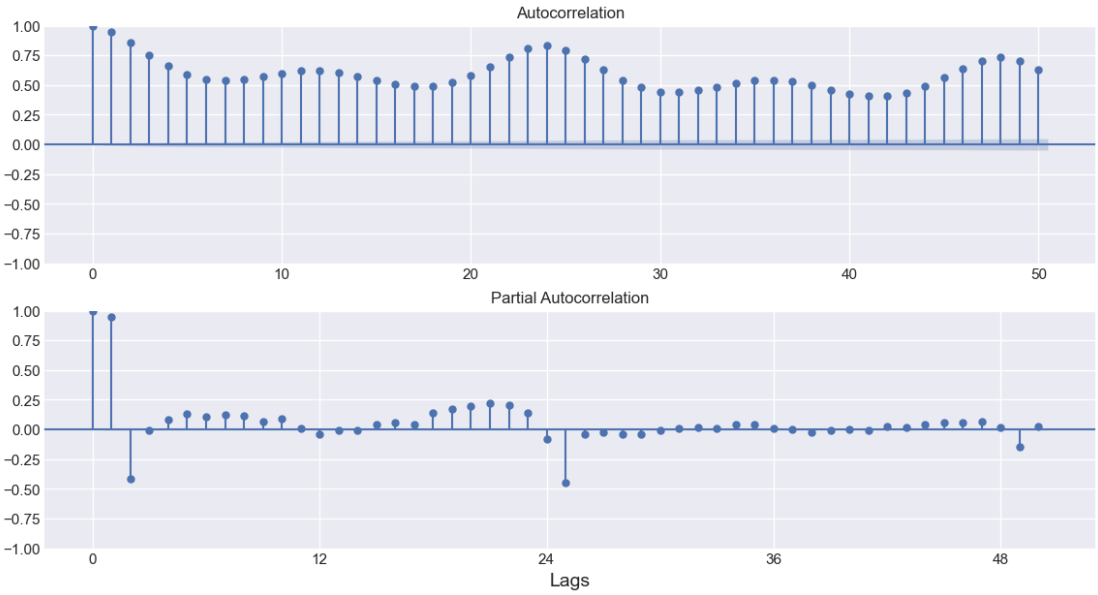

Figura 4.1: ACF e PACF della serie temporale del PUN

La rappresentazione grafica di queste funzioni è essenziale poichè fornisce un'indicazione visiva dei potenziali termini AR e MA da includere nel modello. Ad esempio, un taglio netto dopo un certo numero di lags (ritardi) nel PACF suggerisce un possibile ordine AR, mentre un taglio netto nell'ACF suggerisce un ordine MA.

Il grafico ACF, in Figura 4.1, presenta un declino lento e regolare delle correlazioni, che suggerisce un modello SARIMA con componenti MA significativi. La presenza di autocorrelazioni significative a molti ritardi può indicare anche una forte componente stagionale.
Dal grafico PACF si evince, invece, che a seguito di un picco significativo al primo ritardo i valori si avvicinano rapidamente allo zero e fluttuano all'interno delle bande di confidenza. Questo comportamento è tipico di un processo autoregressivo di primo ordine, AR(1).

4.2.3 Scelta dei Parametri

Attraverso un processo di "search grid" tramite il comando `autoarima`, è stato individuata la combinazione dei parametri ottimale basata sul *Akaike Information Criterion* (AIC).
L'AIC è utilizzato per bilanciare la bontà di adattamento del modello con la complessità del modello stesso. Un modello con un valore AIC inferiore è generalmente preferito, in quanto suggerisce una migliore adattabilità con meno parametri.

I risultati dell'AIC, ripotati in Tabella 4.2, suggeriscono come miglior combinazione quella con un valore AR e MA di secondo ordine, differenziazione nulla e stagionalità giornaliera, ossia:

$$(2, 0, 2, 24)x(2, 0, 2, 24)\ AIC = 242954.20$$

Tabella 4.2: Scelta dei parametri del modello tramite AIC

AIC Results	
ARIMA(0, 0, 0, 24)x(0, 0, 0, 24)24	AIC:497838.5003939959
ARIMA(0, 0, 1, 24)x(0, 0, 1, 24)24	AIC:405505.4963705749
ARIMA(0, 0, 2, 24)x(0, 0, 2, 24)24	AIC:340406.0257098431
ARIMA(0, 1, 0, 24)x(0, 1, 0, 24)24	AIC:306471.4005660789
ARIMA(0, 1, 1, 24)x(0, 1, 1, 24)24	AIC:261173.5005849315
ARIMA(0, 1, 2, 24)x(0, 1, 2, 24)24	AIC:257828.4166792343
ARIMA(0, 2, 0, 24)x(0, 2, 0, 24)24	AIC:381301.9489247888
ARIMA(0, 2, 1, 24)x(0, 2, 1, 24)24	AIC:306417.3905535026
ARIMA(0, 2, 2, 24)x(0, 2, 2, 24)24	AIC:272684.50119439955
ARIMA(1, 0, 0, 24)x(1, 0, 0, 24)24	AIC:281514.35538581293
ARIMA(1, 0, 1, 24)x(1, 0, 1, 24)24	AIC:258645.78535601668
ARIMA(1, 0, 2, 24)x(1, 0, 2, 24)24	AIC:255540.272162497
ARIMA(1, 1, 0, 24)x(1, 1, 0, 24)24	AIC:259339.4129797056
ARIMA(1, 1, 1, 24)x(1, 1, 1, 24)24	AIC:254933.08704140937
ARIMA(1, 1, 2, 24)x(1, 1, 2, 24)24	AIC:244378.91168977995
ARIMA(1, 2, 0, 24)x(1, 2, 0, 24)24	AIC:313590.037072713
ARIMA(1, 2, 1, 24)x(1, 2, 1, 24)24	AIC:259409.4976598138
ARIMA(1, 2, 2, 24)x(1, 2, 2, 24)24	AIC:255099.25063923866
ARIMA(2, 0, 0, 24)x(2, 0, 0, 24)24	AIC:254548.13072642472
ARIMA(2, 0, 1, 24)x(2, 0, 1, 24)24	AIC:252777.30473922435
ARIMA(2, 0, 2, 24)x(2, 0, 2, 24)24	**AIC:242954.20349145515**
ARIMA(2, 1, 0, 24)x(2, 1, 0, 24)24	AIC:258139.62683271238
ARIMA(2, 1, 1, 24)x(2, 1, 1, 24)24	AIC:246998.66668018242
ARIMA(2, 1, 2, 24)x(2, 1, 2, 24)24	AIC:244387.63899724616
ARIMA(2, 2, 0, 24)x(2, 2, 0, 24)24	AIC:301111.8041761193
ARIMA(2, 2, 1, 24)x(2, 2, 1, 24)24	AIC:258212.06427305413

Best SARIMAX parameters: (2, 0, 2, 24)x(2, 0, 2, 24)24 with AIC: 242954.20349145515

4.2.4 Addestramento del Modello

Una volta identificata la migliore combinazione, tramite il comando `model.fit`, il modello è stato adattato al training set. I risultati vengono riportati in Tabella 4.3.

Tabella 4.3: Addestramento del modello SARIMA sui dati di training

SARIMA Results

Dep. Variable:		PUN		No. Observations:	46005
Model:	SARIMA (2, 0, 2)x(2, 0, 2, 24)			Log Likelihood	-121468.102
Date:	Mon, 27 Nov 2023			AIC	242954.203
Time	19:37:32			BIC	243032.832
Sample:	0			HQIC	242978.929
	- 46005				
Covariance Type:	opg				
	Coeff.	Std Err	z	P>|z|	[0.025 0.975]
ar.L1	1.8184	0.006	291.177	0.000	1.806 1.831
ar.L2	-0.8212	0.006	-138.030	0.000	-0.833 -0.810
ma.L1	-0.8236	0.007	-125.424	0.000	-0.837 -0.811
ma.L2	-0.1174	0.003	-37.024	0.000	-0.124 -0.111
ar.S.L24	1.0767	0.017	62.270	0.000	1.043 1.111
ar.S.L48	-0.0809	0.017	-4.714	0.000	-0.115 -0.047
ma.S.L24	-0.7748	0.017	-44.664	0.000	-0.809 -0.741
ma.S.L48	-0.1094	0.015	-7.440	0.000	-0.138 -0.081
sigma2	11.5098	0.033	353.900	0.000	11.446 11.574
Ljung-Box (L1) (Q):		0.06		Jarque-Bera (JB):	195038.39
Prob(Q):		0.82		Prob(JB):	0.00
Heteroskedasticity (H):		0.68		Skew:	0.12
Prob(H) (two-sided):		0.00		Kurtosis:	13.38

Diagnostica dei Residui

Al fine di valutare la bontà dell'adattamento di un modello SARIMA è buona norma analizzare i *Residui*. Essi, infatti, rappresentano gli errori di previsione, ossia le differenze tra i valori osservati y_t e i valori previsti $\hat{y}_t$ dal modello calcolati come segue:

$$e_t = y_t - \hat{y}_t$$

dove:

- e_t è il residuo al tempo t;
- y_t è il valore osservato al tempo t;
- $\hat{y}_t$ è il valore previsto dal modello SARIMA al tempo t.

L'analisi dei residui è un passaggio cruciale nella valutazione di un modello SARIMA. Se i residui non sembrano essere rumore bianco (cioè, non mostrano pattern, autocorrelazione o non-normalità significativi), ciò può indicare che il modello non risulta adeguato e potrebbe essere migliorato. L'analisi, inoltre, permette di rivelare pattern che il modello non è riuscito a catturare.

In un modello ben adattato, dunque, i residui dovrebbero essere:

- **Indipendenti**: Non mostrano autocorrelazione;
- **Distribuiti normalmente**: Preferibilmente con una media vicino a zero;
- **Omogenei**: Hanno una varianza costante nel tempo (*omoscedasticità*).

Sulla base dei risultati dell'addestramento riportati in Tabella 4.3, è possibile trarre le seguenti considerazioni:

1. Ljung-Box (Q):
    - Statistic (Q): 0.06 è il valore della statistica Q del test di Ljung-Box, che verifica l'assenza di autocorrelazione nei residui a diversi lag. Un valore basso suggerisce che non ci sono prove contro l'ipotesi nulla di assenza di autocorrelazione.

- Prob(Q): 0.82 è il p-value associato al test di Ljung-Box. Un valore alto (maggiore di 0.05 di solito indica un livello di significatività del 5%) suggerisce che non si rifiuta l'ipotesi nulla e che i residui non mostrano autocorrelazione significativa.

2. Jarque-Bera (JB):
   - Statistic (JB): 195038.39 è una statistica molto elevata per il test di Jarque-Bera, che verifica la normalità dei residui basandosi su skewness e kurtosis. Un valore così alto indica una forte deviazione dalla normalità.
   - Prob(JB): 0.00 (o un valore molto vicino a zero) è il p-value del test di Jarque-Bera. Un p-value vicino a zero indica il rifiuto dell'ipotesi nulla e conferma che i residui non sono normalmente distribuiti.
3. Heteroskedasticity (H):
   - H: 0.68 è il valore della statistica del test di eteroschedasticità, che verifica se la varianza dei residui è costante nel tempo. Un valore minore di 1 potrebbe suggerire la presenza di eteroschedasticità, ossia che la varianza dei residui cambia nel tempo.
   - Prob(H) (two-sided): 0.00 è il p-value associato al test di eteroschedasticità. Un p-value così basso suggerisce che si rifiuta l'ipotesi di omoschedasticità, e quindi, viene confermata l'ipotesi di eteroschedasticità
4. Skewness: 0.12 indica un leggero "skew" (asimmetria) nei residui. Un valore vicino a zero suggerisce che i residui hanno una distribuzione simmetrica attorno alla media. Tuttavia, in presenza di un'elevata kurtosis, questo valore potrebbe non essere molto informativo.
5. Kurtosis: 13.38 indica una kurtosis molto alta. Una kurtosis maggiore di 3 (che rappresenta la kurtosis di una distribuzione normale) suggerisce una distribuzione "leptocurtica", che significa che i dati hanno code più pesanti e un picco più acuto rispetto a una distribuzione normale.

ACF e PACF dei Residui Il grafico di autocorrelazione e autocorrelazione parziale dei residui aiuta a verificare l'assenza di dipendenza temporale interna tra i residui stessi. Se il modello ha catturato tutte le informazioni significative nella serie temporale, i residui non dovrebbero mostrare autocorrelazione. Da entrambi i grafici raffigurati in Figura 4.2 e Figura 4.3 si osserva che la maggior parte dei coefficienti di autocorrelazione e autocorrelazione parziale sono vicini a zero e rientrano entro i limiti di confidenza (indicati dalle linee blu orizzontali), il che suggerisce che non c'è autocorrelazione significativa nei residui.

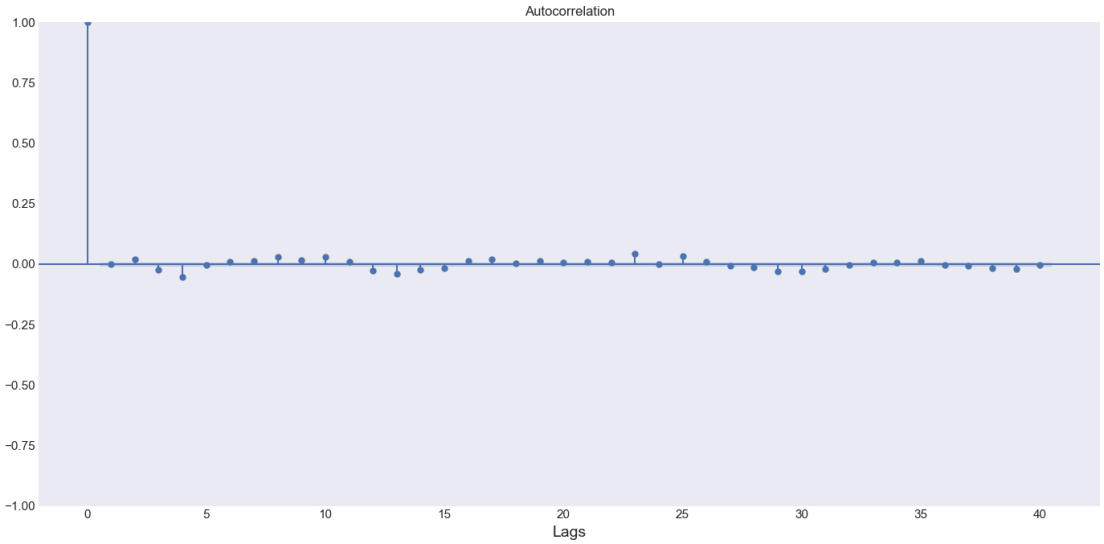

Figura 4.2: AC Function dei residui

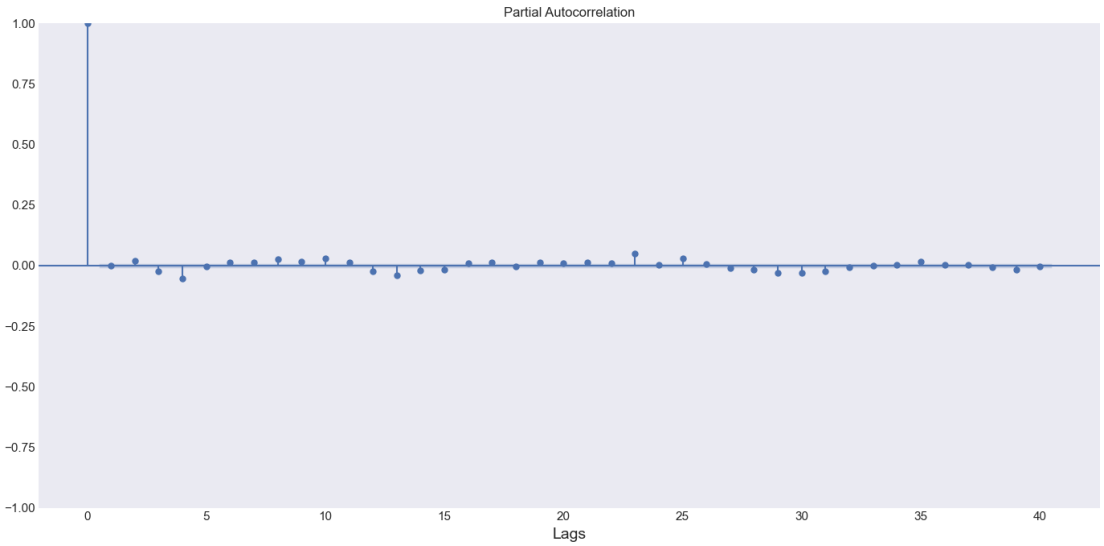

Figura 4.3: PAC Function dei residui

I risultati dell'addestramento, dunque, suggeriscono che nonostante i residui non mostrino autocorrelazione significativa confermando l'ipotesi di indipendenza interna, essi non risultano ne seguire una distribuzione normale, ne confermare l'ipotesi di omoschedasticità; ciò potrebbe influenzare la validità del modello dato che i modelli autoregressivi presuppongono la normalità e l'omoschedasticità dei residui.

4.2.5 Validazione del Modello

Il modello è stato successivamente valutato sul set di test e le previsioni calcolate sono state confrontate con i valori reali (Figura 4.4 e Figura 4.5).

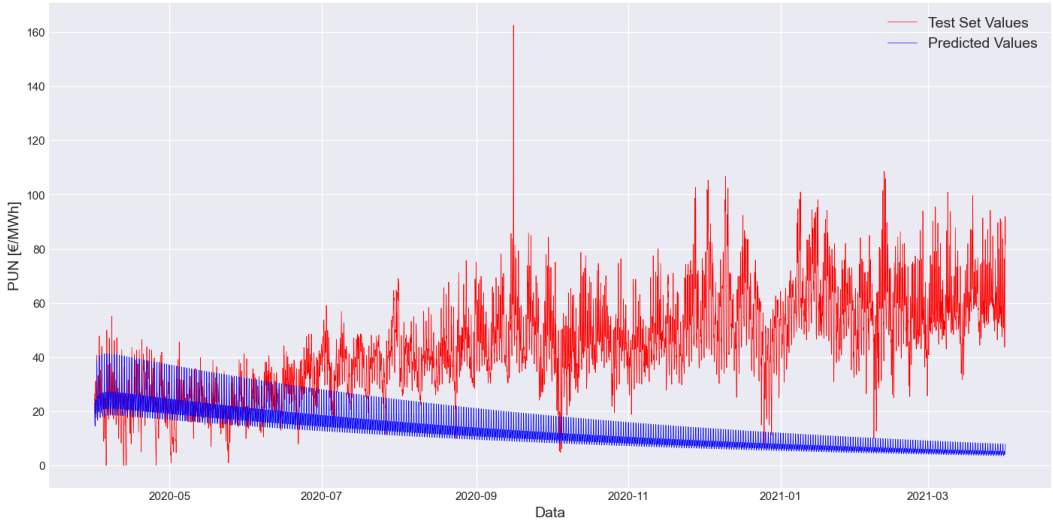

Figura 4.4: Validazione del modello e confronto tra valori reali e valori predetti

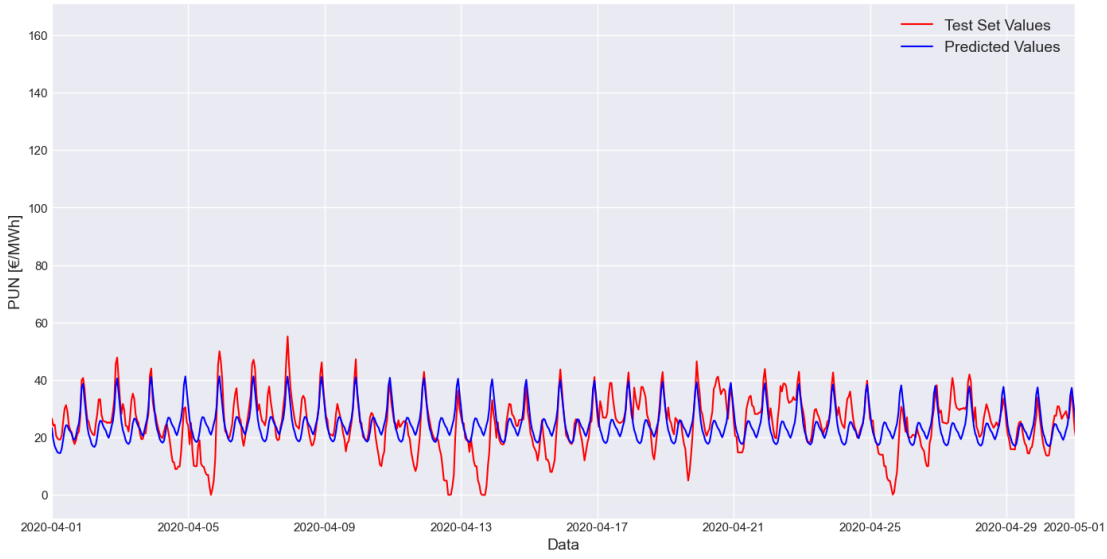

Figura 4.5: Validazione del modello e confronto tra valori reali e valori predetti - focus mensile

Come si evinceva dall'analisi dei risultati dell'addestramento risulta evidente come il modello SARIMA non riesce a catturare la variabilità dei dati di test (in rosso) tendendo a produrre previsioni notevomente sottostimate che seguono un andamento piuttosto discordante.
Le ragioni che potrebbero spiegare questo comportamento sono le seguenti:

1. **Underfitting:** il modello potrebbe essere troppo poco complesso e non avere abbastanza capacità per catturare la complessità dei dati. L'underfitting, infatti, si verifica quando il modello non riesce a catturare la varianza dei dati di addestramento portando, di conseguenza, ad una scarsa capacità di generalizzare i dati di test;

2. **Mancanza di caratteristiche rilevanti:** il modello SARIMA, per natura, non permette di considerare l'influenza che altre variabili esterne possano avere sulla serie target. A tal proposito, sarebbe opportuno rendere il modello più complesso considerando delle variabili esogene [1].

4.2.6 Valutazione della Performance

Nella valutazione delle prestazioni dei modelli di previsione, specialmente quelli applicati alle serie temporali, è cruciale utilizzare metriche appropriate che riflettano accuratamente la bontà di adattamento e l'accuratezza delle previsioni. Le metriche utilizzate sono: il *Root Mean Square Error* (RMSE) e l'*Errore Assoluto Medio* (MAE).

- **Root Mean Square Error (RMSE):** l'RMSE quantifica l'errore medio dei valori previsti rispetto ai valori osservati. Un RMSE di zero indica che il modello ha previsto perfettamente i valori senza alcun errore.
  È definito come segue:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

[1] Modelli autoregressivi stagionali che tengono conto anche delle variabili esterne vengono chiamati SARIMAX.

dove y_i sono i valori osservati, $\hat{y}_i$ sono i valori previsti, e n è il numero totale di osservazioni.
Per il modello SARIMA, l'RMSE calcolato è pari a **37,657**.

- **Mean Absolute Error (MAE):** il MAE misura la media degli errori assoluti tra i valori osservati e quelli previsti, fornendo una misura più diretta dell'errore medio.
  È definito come segue:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

  Per il modello SARIMA, il MAE calcolato è pari a **31,721** suggerendo che, in media, le previsioni del modello differiscono dai valori reali di circa 31,721 unità.

Tabella 4.4: Metriche di Performance - SARIMA

Valutazione dell'errore	
RMSE	37,657
MAE	31,721

4.3 Conclusioni

Dai risultati finora analizzati, emerge chiaramente che il modello SARIMA non si adatta efficacemente ai dati in esame, rendendolo inadeguato per le previsioni future. Sebbene ci sia la possibilità di ottimizzarlo ulteriormente, come affermato in precedenza, si è deciso di esplorare un approccio differente attraverso l'adozione di un modello più sofisticato che tenga conto dell'influenza di variabili indipendenti esterne: la *Regressione Lineare Multipla*. Quest'ultima verrà esaminata in dettaglio nel prossimo capitolo.

Capitolo 5

Regressione Lineare Multipla

La Regressione Lineare rappresenta uno dei metodi fondamentali e ampiamente utilizzati nell'ambito dell'analisi statistica. Questa tecnica, attraverso l'uso di un modello matematico lineare, mira a descrivere la relazione tra una variabile dipendente e una variabile indipendente (Regressione Lineare Semplice, Figura 5.1) o più variabili indipendenti (Regressione Lineare Multipla, Figura 5.2).
Le variabili possono essere suddivise in:

- Predittori: sono le variabili da cui il modello estrarrà le relazioni per stimare l'output.
- Target: sono le variabili che si intende stimare attraverso il modello.

L'obiettivo della regressione lineare è quello di prevedere i valori della variabile target come combinazione lineare dei predittori minimizzando una funzione di errore. Il problema principale, dunque, che si affronta nella costruzione di un modello di questo tipo è quello di trovare la retta che presenta il miglior adattamento (*best-fit*) ai punti osservati.

5.1 Architettura Matematica

L'architettura matematica della regressione lineare è la seguente:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \epsilon$$

in cui:

- Y è la variabile dipendente (target);
- X_i sono le variabili indipendenti (predittori);
- β_0 l'intercetta sull'asse Y;
- β_i, $i \neq 0$ i coefficienti di regressione parziale: indicano di quanto varia in media Y quando X_i aumenta di un'unità, a parità di valori delle altre variabili indipendenti;
- ϵ è il termine di errore che rappresenta la discrepanza tra i valori osservati e quelli previsti dalla linea di regressione: traduce l'incapacità del modello di riprodurre con esattezza la realtà osservata;

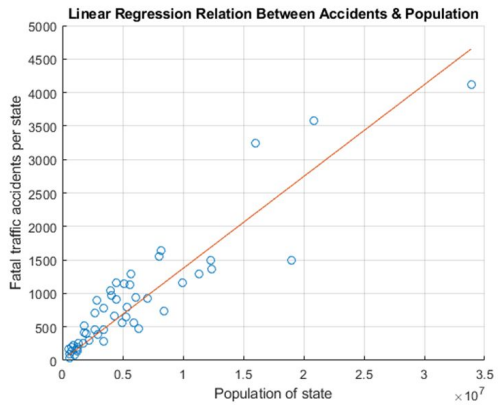

Figura 5.1: Esempio di regressione lineare semplice (Fonte:MathWorks)

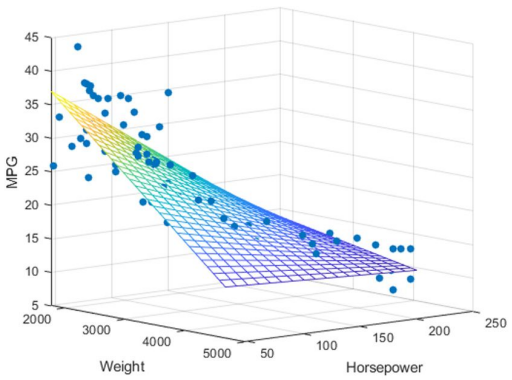

Figura 5.2: Esempio di regressione lineare multipla (Fonte:MathWorks)

5.1.1 Stima dei Coefficienti

La stima dei coefficienti β_0 (intercetta) e β_i (coefficienti di regressione parziale) avviene attraverso il *Metodo dei Minimi Quadrati* (Ordinary Least Squares, OLS).

Il metodo OLS si basa sul principio di minimizzare la somma dei quadrati delle differenze tra i valori osservati e quelli previsti dal modello e si pone l'obiettivo di trovare i valori dei coefficienti di regressione parziale (β_i) che minimizzano la *funzione di costo*, definita come la somma dei quadrati dei residui (RSS, Residual Sum of Squares):

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

dove y_i sono i valori osservati della variabile dipendente, $\hat{y}_i$ sono i valori previsti dal modello, e n è il numero di osservazioni eseguite.

Per minimizzare l'RSS, viene calcolata la derivata parziale di RSS rispetto a β_i uguale a zero:

$$\frac{\partial RSS}{\partial \beta} = -2X^T(y - X\beta) = 0$$

Da cui si ottiene il valore di β_i:

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

5.2 Processo di Calcolo

Verranno generati due modelli regressivi: il primo modello si concentra sull'uso delle variabili esogene DEM e GAS (Modello A), mentre il secondo modello (Modello B) si propone di esplorare l'eventuale miglioramento della qualità predittiva attraverso l'incorporazione delle variabili SOLAR e WIND, valutando così in maniera diretta l'impatto delle energie rinnovabili sulla precisione delle previsioni. Entrambi i modelli includeranno, inoltre, variabili (dummies) temporali, concepite per catturare gli effetti legati a differenti periodicità: giornaliera, settimanale, mensile e annuale.

I modelli regressivi sono definiti dalle seguenti relazioni:

Modello A

$$PUN_t = \beta_0 + \beta_1 DEM_t + \beta_1 GAS_t + \gamma D_t + \epsilon_t$$

Modello B

$$PUN_t = \beta_0 + \beta_1 DEM_t + \beta_1 GAS_t + \beta_1 SOLAR_t + \beta_1 WIND_t + \gamma D_t + \epsilon_t$$

Prima di procedere con la modellazione verrà condotta una *decomposizione* delle serie temporali. Questo processo permetterà di isolare e esaminare le componenti di *trend*, *stagionalità* e *residui*, step indispensabile non solo per acquisire una comprensione più profonda della struttura intrinseca dei dati ma anche per identificare eventuali anomalie che potrebbero influenzare significativamente le performance dei modelli regressivi.

Successivamente verrà affrontato il problema della non-stazionarietà delle variabili esogene, in particolare quella relativa al GAS, applicando la differenziazione.

Si concluderà il processo addestrando entrambi i modelli sottoponendoli a un confronto diretto tramite l'uso di metriche di valutazione specifiche. Ciò consentirà di determinare se l'introduzione delle variabili SOLAR e WIND abbia effettivamente portato ad un incremento della qualità predittiva del modello di partenza.

5.2.1 Decomposizione e Analisi di Correlazione

La decomposizione delle serie temporali è un'importante tecnica di analisi preliminare utilizzata per identificare e isolare i componenti principali di una serie temporale: trend, stagionalità e residui.

Per eseguire la decomposizione è stato adottato un approccio basato su componenti additive, utilizzando il pacchetto `statsmodels` in Python. In particolare, mediante la funzione `seasonal_decompose` del modulo `tsa.seasonal` è stato possibile scomporre le serie temporali nelle componenti principali, ossia:

1. **Trend:** indica la direzione generale in cui si muove una serie temporale nel lungo periodo. Può essere crescente, decrescente o costante. Il trend, dunque, riflette l'effetto di fattori di lungo periodo sulla variabile di interesse;
2. **Stagionalità:** rappresenta le fluttuazioni periodiche che si ripetono in intervalli regolari, come giornalieri, settimanali, mensili o annuali. Queste variazioni sono spesso legate a fattori stagionali come il clima, le festività o altri cicli economici;
3. **Residui:** componente della serie temporale non spiegata dal trend o dalla stagionalità. Questi possono includere variazioni casuali o irregolarità non catturate da quest'ultime.

Analizzando i risultati della decomposizione (riportati in Appendice A.3) è possibile dedurre le seguenti osservazioni:

- PUN
    1. Il trend mostra valori relativamente stabili nel breve periodo analizzato, indicando che non ci sono stati cambiamenti significativi nel prezzo dell'energia in quelle ore;
    2. La stagionalità evidenzia variazioni negative nelle prime ore del giorno, suggerendo un calo nel prezzo dell'energia durante queste ore, che potrebbe riflettere una minore domanda o un aumento dell'offerta di energia (ad esempio, dalla produzione solare);
    3. I residui mostrano variazioni che potrebbero indicare fattori non catturati dal modello o fluttuazioni casuali.

- DEM

  1. Il trend della domanda di energia mostra una leggera diminuzione nel corso della giornata analizzata.
  2. La stagionalità indica cali significativi durante la notte, il che è atteso data la minore attività e quindi la minore domanda di energia.
  3. I residui suggeriscono piccole variazioni non spiegate dalla stagionalità o dal trend, che potrebbero essere attribuite a eventi specifici o anomalie nella domanda di energia.

- GAS

  1. Il trend mostra valori leggermente decrescenti, indicando una leggera riduzione del prezzo del gas in quelle ore.
  2. La stagionalità è quasi inesistente, il che potrebbe indicare che il prezzo del gas è meno influenzato da fattori stagionali nell'arco della giornata.
  3. I residui mostrano variazioni minime, suggerendo che altri fattori non analizzati potrebbero influenzare il prezzo del gas.

- SOLAR

  1. Il trend è costante, il che potrebbe riflettere una produzione solare stabile in quelle ore.
  2. La stagionalità mostra valori negativi estremamente elevati, il che è spiegato da un'assenza di produzione solare durante la notte (come atteso) e quindi una differenza negativa rispetto al valore medio.
  3. I residui possono riflettere variazioni nella produzione solare non spiegate dal modello di stagionalità, come l'effetto delle condizioni meteorologiche.

- WIND

  1. Il trend è stabile, indicando una produzione costante di energia eolica nelle ore analizzate.
  2. La stagionalità suggerisce variazioni periodiche e irregolarità dimostrando una minore prevedibilità
  3. I residui, al pari della stagionalità, suggeriscono variazioni periodiche e irregolarità che potrebbero essere legate alla variabilità del vento.

Infine, tramite la libreria `scipy.stats` e il comando `pearsonr`, è stata eseguita la *Correlazione di Pearson* (Tabella 5.1), una misura statistica che quantifica

la relazione lineare tra due variabili quantitative. Il *Coefficiente di correlazione*, indicato con r, varia tra -1 e +1, ed è definito come segue:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

dove:

- x_i e y_i sono i valori delle due variabili;
- $\bar{x}$ e $\bar{y}$ sono le medie aritmetiche dei valori di x e y;
- n è il numero di osservazioni;
- $+1$ indica una correlazione perfettamente positiva: se una variabile aumenta, anche l'altra aumenta in modo proporzionale;
- -1 indica una correlazione perfettamente negativa: se una variabile aumenta, l'altra diminuisce in modo proporzionale;
- 0 indica l'assenza di correlazione lineare tra le due variabili: le variazioni di una variabile non sono associate in modo prevedibile alle variazioni dell'altra.

Il calcolo del coefficiente di correlazione di Pearson si basa sulla covarianza tra le due variabili, normalizzata rispetto al prodotto delle loro deviazioni standard. Questo permette di valutare quanto strettamente due variabili tendono a cambiare insieme, rispetto alla loro variabilità individuale. Tuttavia, è importante notare che un valore elevato di r non implica necessariamente una relazione di causa-effetto tra le due variabili. Inoltre, r può non rilevare relazioni non lineari, per le quali potrebbero essere più adatte altre misure di correlazione.

Tabella 5.1: Correlazione di Pearson

Correlazione di Pearson					
	PUN	DEM	SOLAR	WIND	GAS
PUN	1.000000	0.592484	-0.093381	-0.109068	0.117632
DEM	0.592484	1.000000	0.332522	0.002371	0.003623
SOLAR	-0.093381	0.332522	1.000000	-0.064633	-0.035727
WIND	-0.109068	0.002371	-0.064633	1.000000	0.101012
GAS	0.117632	0.003623	-0.035727	0.101012	1.000000

Dal test emerge chiaramente come il PUN sembri dipendere in modo significativo dal fabbisogno elettrico e dal prezzo MGP del gas naturale.

5.2.2 Test della stazionarietà e differenziazione

Stazionarietà Tramite il pacchetto `adfuller` della libreria `statsmodels.tsa.statools` è stato eseguito il test di *Dickey-Fuller aumentato* (ADF) per verificare la stazionarietà delle serie temporali.
Come si evince dai risultati riportati in Tabella 5.2 e Tabella 5.3, per la serie del GAS non è stato possibile rifiutare l'ipotesi nulla di radice unitaria suggerendo dunque una non stazionarietà della stessa.

Tabella 5.2: Test ADF per la stazionarietà - DEM, GAS

Augmented Dickey-Fuller Test	
DEM	
ADF test statistic	-18.978558
p-value	0.000000
lags used	59.000000
observations	54704.000000
critical value (1%)	-3.430470
critical value (5%)	-2.861593
critical value (10%)	-2.566798
Strong evidence against the null hypothesis	
Reject the null hypothesis	
Data has no unit root and is stationary	
GAS	
ADF test statistic	-1.378693
p-value	0.592447
lags used	48.000000
observations	54715.000000
critical value (1%)	-3.430470
critical value (5%)	-2.861593
critical value (10%)	-2.566798
Weak evidence against the null hypothesis	
Fail to reject the null hypothesis	
Data has a unit root and is not stationary	

Tabella 5.3: Test ADF per la stazionarietà - SOLAR, WIND

Augmented Dickey-Fuller Test	
SOLAR	
ADF test statistic	-9.652948e+00
p-value	1.418681e-16
lags used	5.400000e+01
observations	5.470900e+04
critical value (1%)	-3.430470e+00
critical value (5%)	-2.861593e+00
critical value (10%)	-2.566798e+00
Strong evidence against the null hypothesis	
Reject the null hypothesis	
Data has no unit root and is stationary	
WIND	
ADF test statistic	-19.641463
p-value	0.000000
lags used	57.000000
observations	54706.000000
critical value (1%)	-3.430470
critical value (5%)	-2.861593
critical value (10%)	-2.566798
Strong evidence against the null hypothesis	
Reject the null hypothesis	
Data has no unit root and is stationary	

Differenziazione

Nel caso della colonna GAS, è stata eseguita una differenziazione di primo ordine. Questo significa che ogni valore nella serie trasformata è la differenza tra il valore corrente e quello immediatamente precedente nella serie originale. Matematicamente, se si denota con Y_t il valore della serie al tempo t, la serie differenziata Y_t' è data da:

$$Y_t' = Y_t - Y_{t-1}$$

Dopo aver applicato la differenziazione, il risultato del test ADF (Tabella 5.4) ha confermato la stazionarietà anche per la colonna GAS.

Tabella 5.4: Test ADF per la stazionarietà - GAS

Augmented Dickey-Fuller Test	
GAS	
ADF test statistic	-36.699375
p-value	0.000000
lags used	47.000000
observations	54715.000000
critical value (1%)	-3.430470
critical value (5%)	-2.861593
critical value (10%)	-2.566798
Strong evidence against the null hypothesis	
Reject the null hypothesis	
Data has no unit root and is stationary	

5.2.3 Dummies Temporali

Al fine di gestire l'effetto di caratteristiche temporali sui dati, sono state geenrate delle dummies temporali (Tabella 5.5) giornaliere, settimanali, mensili e annuali attraverso il codice riportato in Appendice A.3.

Tabella 5.5: Dummies Temporali

Matrice delle dummies temporali

Data e Ora	PUN	DEM	GAS	SOLAR	WIND	hour	dayofweek	quarter	month	year	dayofyear	dayofmonth	weekofyear
2015-01-01 00:00:00	52.327563	24405.00000	24.516	0.0	2223.0	0	3	1	1	2015	1	1	1
2015-01-01 01:00:00	49.892778	23126.00000	24.516	0.0	2326.0	1	3	1	1	2015	1	1	1
2015-01-01 02:00:00	39.100000	21534.00000	24.516	0.0	2487.0	2	3	1	1	2015	1	1	1
2015-01-01 03:00:00	35.870000	20219.00000	24.516	0.0	2820.0	3	3	1	1	2015	1	1	1
2015-01-01 04:00:00	33.400000	19470.00000	24.516	0.0	3091.0	4	3	1	1	2015	1	1	1
						...							
2021-03-31 19:00:00	55.124920	37914.00100	95.119	8571.0	1012.0	19	2	1	3	2021	90	31	13
2021-03-31 20:00:00	60.900000	41071.00025	95.119	0.0	837.0	20	2	1	3	2021	90	31	13
2021-03-31 21:00:00	67.540000	42255.74950	95.119	0.0	712.0	21	2	1	3	2021	90	31	13
2021-03-31 22:00:00	91.931310	38741.99925	95.119	0.0	711.0	22	2	1	3	2021	90	31	13
2021-03-31 23:00:00	81.599870	34969.74975	95.119	0.0	651.0	23	2	1	3	2021	90	31	13

5.2.4 Addestramento del Modello

Tramite il comando `sm.OLS` della libreria import `statsmodels.api` i modelli sono stati prima addestrati sul training set e successivamente adattati ai dati tramite il comando `model.fit` fornendo i risultati riportati in Tabella 5.6 e in Tabella 5.7.

Tabella 5.6: Addestramento del Modello A

Regressione Lineare Multipla (DEM, GAS, Dummies)					
Dep. Variable:	PUN			R-squared:	0.369
Model:	OLS			Adj. R-squared:	0.369
Method:	Least Squares			F-statistic:	1.347e+04
Date:	Thu, 29 Feb 2024			Prob (F-statistic):	0.00
Time:	19:26:30			Log-Likelihood:	-1.8050e+05
No. Observations:	46005			AIC:	3.610e+05
Df Residuals:	46002			BIC:	3.610e+05
Df Model:	2				
Covariance Type:	nonrobust				
	coef	std err	t	P>|t|	[0.025 0.975]
const	6.6397	0.313	21.189	0.000	6.026 7.254
DEM	1.2066	0.008	160.612	0.000	1.192 1.221
GAS	0.3068	0.011	27.675	0.000	0.285 0.329
Omnibus:	5580.881			Durbin-Watson:	0.116
Prob(Omnibus):	0.000			Jarque-Bera (JB):	11987.256
Skew:	0.749			Prob(JB):	0.00
Kurtosis:	5.003			Cond. No.	2.09e+04

Tabella 5.7: Addestramento del Modello B

Regressione Lineare Multipla (DEM, GAS, SOLAR, WIND, Dummies)					
Dep. Variable:	PUN			R-squared:	0.481
Model:	OLS			Adj. R-squared:	0.481
Method:	Least Squares			F-statistic:	1.067e+04
Date:	Thu, 29 Feb 2024			Prob (F-statistic):	0.00
Time:	19:26:30			Log-Likelihood:	-1.7601e+05
No. Observations:	46005			AIC:	3.520e+05
Df Residuals:	46002			BIC:	3.521e+05
Df Model:	4				
Covariance Type:	nonrobust				
	coef	std err	t	P>|t|	[0.025 0.975]
const	6.4354	0.294	21.903	0.000	5.860 7.011
DEM	1.4382	0.007	197.935	0.000	1.424 1.452
GAS	0.2726	0.010	27.092	0.000	0.253 0.292
WIND	-0.0016	3.5e-05	-46.735	0.000	-0.002 -0.002
SOLAR	-0.0017	1.88e-05	-90.909	0.000	-0.002 -0.002
Omnibus:	6916.504			Durbin-Watson:	0.126
Prob(Omnibus):	0.000			Jarque-Bera (JB):	16769.497
Skew:	0.860			Prob(JB):	0.00
Kurtosis:	5.406			Cond. No.	2.19e+04

I risultati dell'addestramento del Modello A (Modello B) rivela diversi punti chiave su cui è possibile effettuare delle considerazioni:

1. L'F-statistic è significativamente alto e il Prob (F-statistic) è 0.00, indicando che il modello nel suo complesso è statisticamente significativo.

2. Coefficienti di regressione:

   - DEM: Il coefficiente per DEM è 1.2066 (1.4382) con un p-value di 0.000, indicando che c'è una forte relazione positiva tra la domanda di energia e PUN. Per ogni unità di aumento in DEM, PUN aumenta di 1.2066 (1.4382) unità, tenendo costante il prezzo del gas.
   - GAS: Il coefficiente per GAS è 0.3068 (0.2726) anch'esso con un p-value di 0.000, mostrando una relazione positiva significativa tra il prezzo del gas e PUN. Questo significa che per ogni unità di aumento nel prezzo del gas, PUN aumenta di 0.3068 (0.2726) unità, a parità di domanda di energia.

- WIND e SOLAR: Entrambi hanno coefficienti negativi (-0.0016 per WIND e -0.0017 per SOLAR). La relazione negativa tra PUN e la produzione di energia da fonti rinnovabili riflette l'effetto di abbassamento dei prezzi che la produzione rinnovabile può avere sul mercato dell'energia elettrica.

3. Statistiche diagnostiche dei residui:

   - Durbin-Watson: il valore di Durbin-Watson è 0.116 (0.126), suggerendo la presenza di autocorrelazione positiva tra i residui. Questo potrebbe indicare che il modello non cattura completamente la dinamica temporale tra le variabili.
   - Jarque-Bera (JB): 11987.256 (16769.497) è una statistica molto elevata per il test di Jarque-Bera, che verifica la normalità dei residui basandosi su skewness e kurtosis. Un valore così alto indica una forte deviazione dalla normalità, condizione confermata da p-value del test di Jarque-Bera. Un p-value vicino a zero indica il rifiuto dell'ipotesi nulla e conferma che i residui non sono normalmente distribuiti.
   - Cond. No.: è un parametro che indica la multicollinearità, ossia un fenomeno che si verifica nei modelli di regressione lineare multipla quando una o più delle variabili indipendenti risultano fortemente correlate tra loro. Il numero di condizione è 2.09e+04 (2.19e+04), suggerendo che il modello potrebbe soffrire di multicollinearità.
   - Omnibus: è un test statistico che misura la deviazione dei residui dalla distribuzione normale. Il valore di Omnibus è 5580.881 (6916.504), il quale è significativamente alto, indicando che i residui si discostano in modo significativo dalla normalità.
   - Skew: misura l'asimmetria della distribuzione dei residui rispetto alla media. Un valore di 0 indica una distribuzione simmetrica.Il valore di skew è 0.749 (0.860), suggerendo una distribuzione dei residui moderatamente asimmetrica, inclinata verso destra.
   - Kurtosis: valuta la "pesantezza" delle code della distribuzione dei residui. Una kurtosis maggiore di 3 (che rappresenta la kurtosis di una distribuzione normale) suggerisce una distribuzione "leptocurtica", che significa che i dati hanno code più pesanti e un picco più acuto rispetto a una distribuzione normale.

La mancata normalità dei residui potrebbe compromettere l'affidabilità suggerendo che il modello potrebbe essere migliorato ulteriormente.

5.2.5 Validazione del Modello

Il modello è stato successivamente valutato sul set di test e le previsioni calcolate sono state confrontate con i valori reali, riportando i risultati in Figura 5.3. Per una maggiore comprensione, le previsioni sono state rappresentate su una finestra temporale mensile (Figura 5.4).
Come si evince dai grafici, entrambi i modelli regressivi tendono a sovrastimare i valori del PUN sebbene il Modello B riesca ad essere più accurato del Modello A.

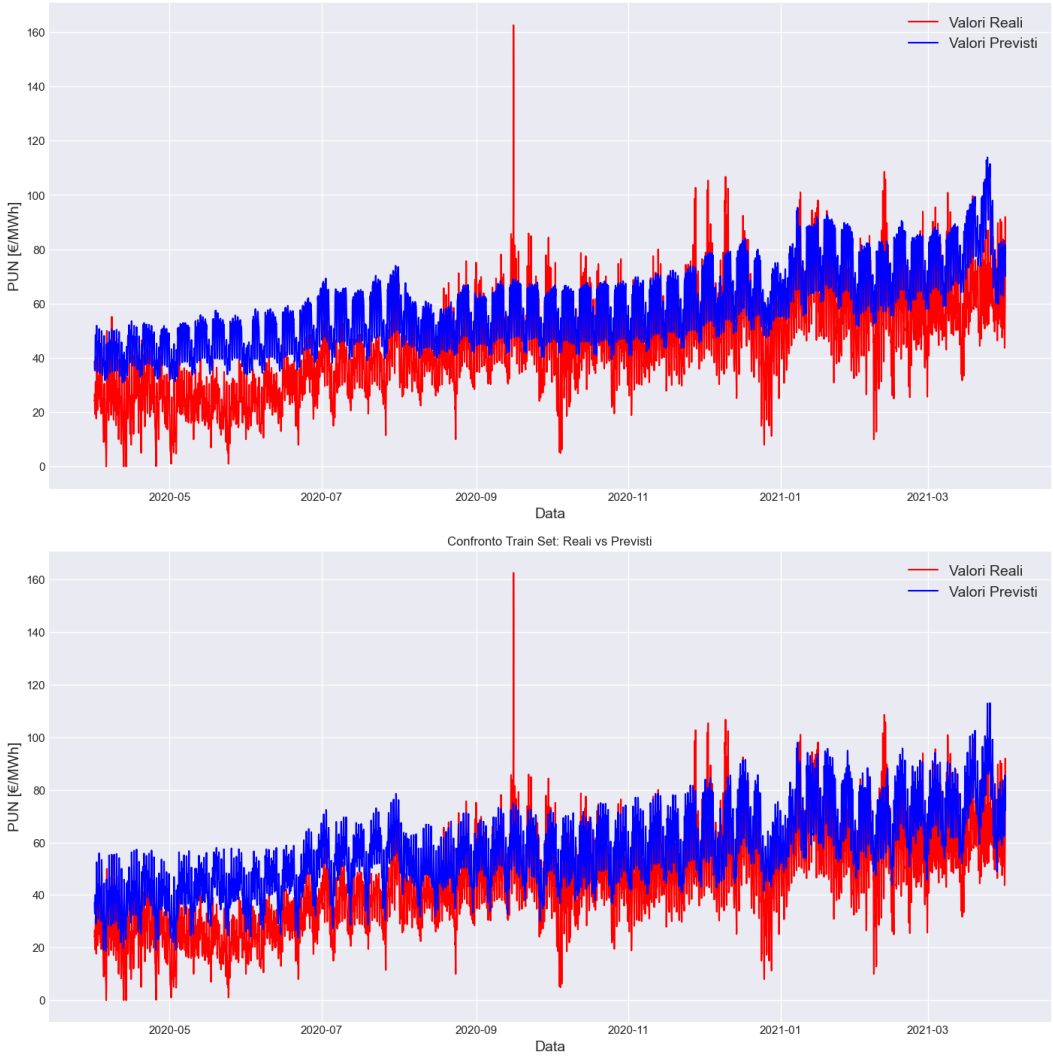

Figura 5.3: Validazione del Modello A (sopra) e del Modello B (sotto) - confronto tra valori reali e valori predetti

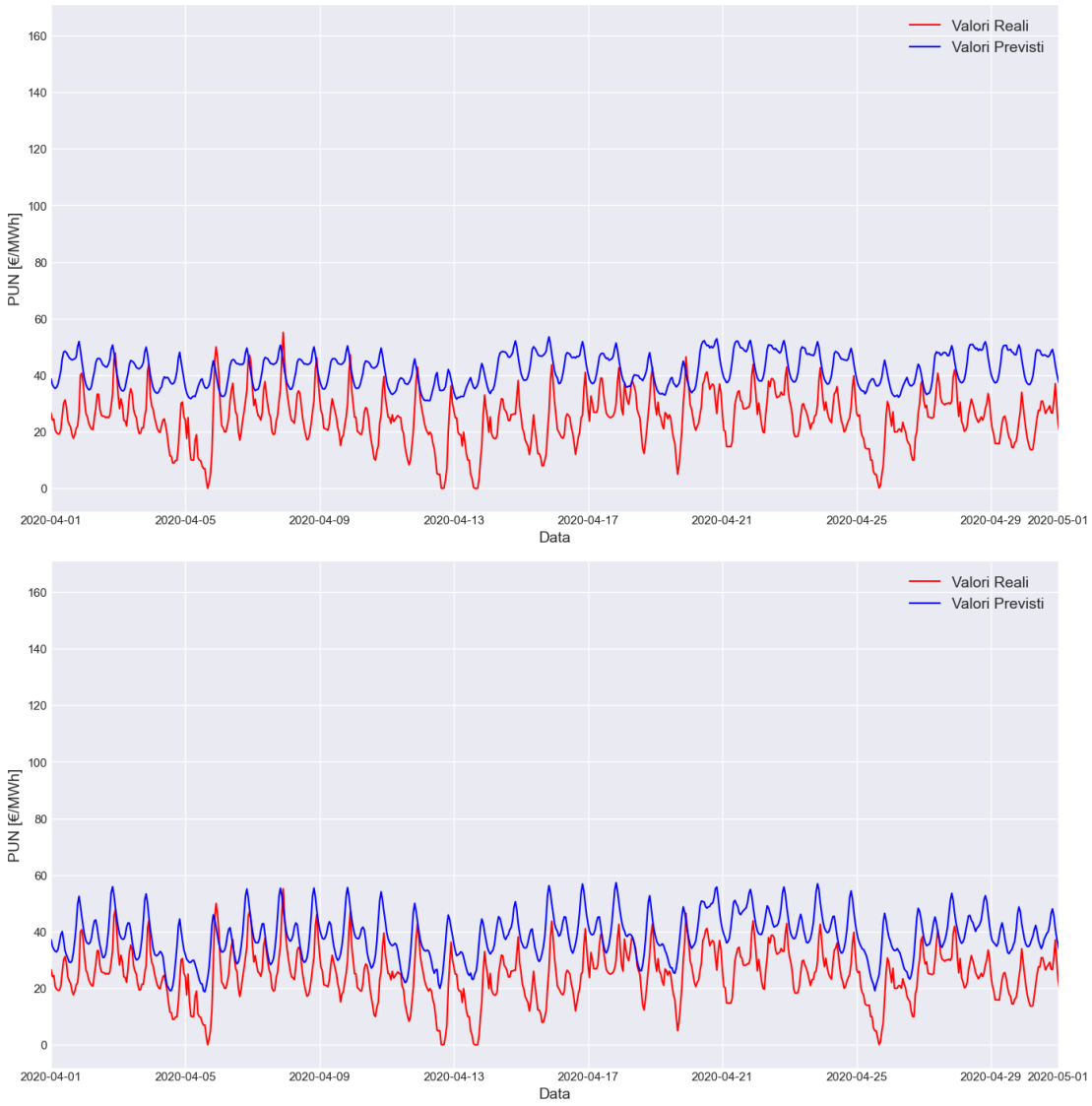

Figura 5.4: Validazione del Modello A (sopra) e del Modello B (sotto) - confronto tra valori reali e valori predetti (focus mensile)

5.2.6 Valutazione della Performance

Nella valutazione delle prestazioni dei modelli di regressione, oltre ad utilizzare metriche di valutazione come MAE e RMSE è cruciale utilizzare il coefficiente di determinazione, noto come R^2. Il Coefficiente di determinazione è una misura statistica che rappresenta la proporzione della varianza per una variabile dipendente che è spiegata da una o più variabili indipendenti in un modello di regressione. In altre parole, indica quanto bene i valori previsti dal modello si adattano ai dati reali. L'R-squared varia da 0 a 1, dove un R^2 di 0 indica che il modello non spiega affatto la variabilità dei dati attorno alla loro media, mentre un R^2 di 1 indica che il modello spiega perfettamente la variabilità dei dati attorno alla loro media.

L'R-squared, matematicamente, può essere espresso come:

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

- Coefficiente di determinazione: il valore di R-squared del Modello A è **0,369**, indicando che circa il *36.9%* della varianza del PUN è spiegata dal modello. Il valore di R-squared del Modello B è **0,481** , indicando che circa il *48.1%* della varianza del PUN è spiegata dal modello. Questo rappresenta un miglioramento rispetto al Modello A, suggerendo che l'introduzione delle variabili SOLAR e WIND, oltre alle dummies temporali, contribuisce significativamente alla comprensione delle fluttuazioni della variabile target;
- Root Mean Square Error (RMSE): per il Modello A (Modello B), l'RMSE calcolato è pari a **18,53** (**16,38**).
- Mean Absolute Error (MAE): Per il Modello A (Modello B), il MAE calcolato è pari a **15,921** (**13,923**) suggerendo che, in media, le previsioni del modello differiscono dai valori reali di circa 15,921 (13,923) unità.

5.3 Conclusioni

In conclusione, il Modello B, con l'introduzione delle variabili SOLAR e WIND offre una base migliore per la previsione del PUN rispetto al Modello A. Tuttavia, come è risultato evidente, l'applicabilità e l'efficacia degli stimatori OLS possono essere influenzate da vari fattori, tra cui l'eteroschedasticità e la multicollinearità, i quali suggeriscono la necessità di ulteriori indagini per ottimizzare il modello. Ciò nonostante, si è optato per esplorare il problema con un approccio differente, ossia attraverso l'adozione di un modello di insieme (Modello Ensemble) in grado di combinare le previsioni di più modelli regressivi di base al fine di produrre un'unica previsione finale generalmente più accurata.

Tabella 5.8: Metriche di Performance - Regressione lineare

Valutazione dell'errore		
	Modello A	Modello B
RMSE	18,530	16,380
MAE	15,921	13,923
R2	0,369	0,481

Capitolo 6

Extreme Gradient Boosting

L'Extreme Gradient Boosting fa parte della famiglia dei modelli Ensemble (Modelli di Insieme) ed è un implementazione del modello *Gradient boosting*.
I modelli Ensemble sono tecniche di Machine Learning che combinano le previsioni di più modelli base al fine di produrre un'unica previsione più accurata rispetto a quella che potrebbe essere ottenuta dai singoli modelli. In particolare, l'algoritmo di Gradient Boosting o Tree Boosting, è una tecnica di apprendimento automatico che si comporta bene se applicata ai modelli regressivi o di classificazione.[13]

6.1 Architettura Matematica dei Modelli Ensemble

L'architettura matematica dietro ai modelli Ensemble può variare a seconda del metodo specifico utilizzato. In particolare, la maggior parte dei modelli Ensemble segue uno dei seguenti principi fondamentali:

- Metodi Averaging: la previsione finale viene calcolata come media delle previsioni dei modelli base, ossia:

$$\hat{y} = \frac{1}{N} \sum_{i=1}^{N} \hat{y}_i$$

  dove:

  - $\hat{y}$ rappresenta la previsione finale ottenuta dalla media delle previsioni;
  - N è il numero totale di modelli;
  - $\hat{y}_i$ è la previsione del i-esimo modello.

- Metodi Boosting: i modelli base sono costruiti sequenzialmente e ciascuno di essi è addestrato per correggere gli errori commessi dal modello precedente. La previsione finale è una somma ponderata delle previsioni dei singoli modelli, ossia [14]

$$\hat{y} = \sum_{i=1}^{N} \alpha_i \hat{y}_i$$

  dove:

  - $\hat{y}$ è la previsione finale ottenuta dalla somma ponderata;
  - α_i rappresenta il peso del i-esimo modello;
  - $\hat{y}_i$ è la previsione del i-esimo modello.

6.2 I Modelli Boosting

Nei capitoli introduttivi sono state presentate le principali tipologie di modelli Ensemble. Nello specifico, verranno approfonditi i modelli *Boosting*.

Ciò che rende tali modelli particolarmente efficaci è la loro capacità di concentrarsi sugli aspetti più difficili da prevedere, assegnando loro progressivamente un peso maggiore nel processo di addestramento.

Esistono vari algoritmi Boosting, ciascuno con le proprie peculiarità:

- **AdaBoost (Adaptive Boosting):** corregge iterativamente i pesi assegnati ai pattern della serie temporale, dando maggiore importanza a quelli erroneamente classificati dai modelli precedenti;
- **XGBoost (Extreme Gradient Boosting):** diversi alberi decisionali vengono aggiunti sequenzialmente al modello nel quale ogni nuovo albero cerca di correggere gli errori commessi da quello precedente. Il processo si basa su un approccio chiamato Gradient Boosting, dove per "gradiente" si intende la derivata della funzione di perdita che l'algoritmo cerca di minimizzare. Il XGBoost può essere impiegato sia nei problemi di regressione che nei problemi di classificazione;
- **CatBoost:** specificamente ottimizzato per gestire variabili categoriche, esso può essere utilizzato senza dover eseguire un'ampia pre-elaborazione dei dati per convertire le categorie in numeri, offrendo allo stesso tempo prestazioni di alto livello.

In questo capitolo verrà approfondito il processo di calcolo eseguito per implementare il modello XGBoost nell'analisi predittiva con lo scopo di migliorare l'accuratezza della previsione eseguita dal modello di Regressione Lineare Multipla affrontato precedentemente.

6.3 Processo di Calcolo

Una peculiarità del modello XGBoost è quella che, se impiegato in analisi di previsione di serie temporali, richiede che esse vengano prima formulate come apprendimento supervisionato. In altre parole, il modello richiede di essere addestrato su un set di dati con caratteristiche di input (*features*) e output (*targets*) ben definite.
In questo caso, la variabile target sarà quella da prevedere, ossia il PUN, mentre le features saranno le variabili temporali, le variabili esogene (DEM ,GAS, WIND e SOLAR) e i *Lag temporali*, ossia features che rappresentano i valori del PUN in punti temporali precedenti. Quest'ultimi sono stati introdotti per catturare le relazioni temporali intrinseche nei dati.

6.3.1 Walk-forward Validation

Una volta terminato il pre-processamento dei dati, il primo passo è stato quello di attuare la *Walk-forward Validation*, ossia una tecnica di validazione che a differenza di quella incrociata (Cross Validation) suddivide i dati in un insieme di addestramento e uno di test rispettando la sequenzialità di quest'ultimi.

In particolare, la validazione Walk-forward, implementata mediante il pacchetto `TimeSeriesSplit` della libreria `scikit-learn` è un processo iterativo che divide il data-set in k parti (folds) in modo sequenziale, assicurando che il set di addestramento preceda temporalmente il set di test per ogni fold. Questo assicura che, per ogni fase di validazione, il modello sia addestrato su dati strettamente antecedenti a quelli su cui verrà testato, rispettando di conseguenza la sequenzialità intrinseca delle serie temporali. In altre parole, dopo aver completato la valutazione su un fold, il set di test precedentemente utilizzato viene integrato nel set di addestramento per la successiva iterazione.

Come riportato nello script in Appendice A.4, la validazione Walk-forward è stata implemetata nel modo seguente:

- **n splits=5:** indica il numero di suddivisioni (o folds) da creare. In questo caso, il data-set sarà suddiviso in 5 parti;
- **test size=12*364*1:** specifica la dimensione del test set per ogni fold, che in questo caso è impostata a 12 mesi (assumendo 364 giorni per anno). Ciò significa che ogni test set conterrà i dati di un anno intero;

- **gap=24:** indica un intervallo di tempo tra il set di addestramento e il set di test per evitare la perdita di informazioni dal futuro al passato. In questo esempio, il gap è di 24 ore, il che assicura che ci sia almeno un giorno tra l'ultimo punto del set di addestramento e il primo punto del set di test in ogni fold.

In Figura 6.1 viene rappresentata la suddivisione tra dati di addestramento e test per ciascun fold.

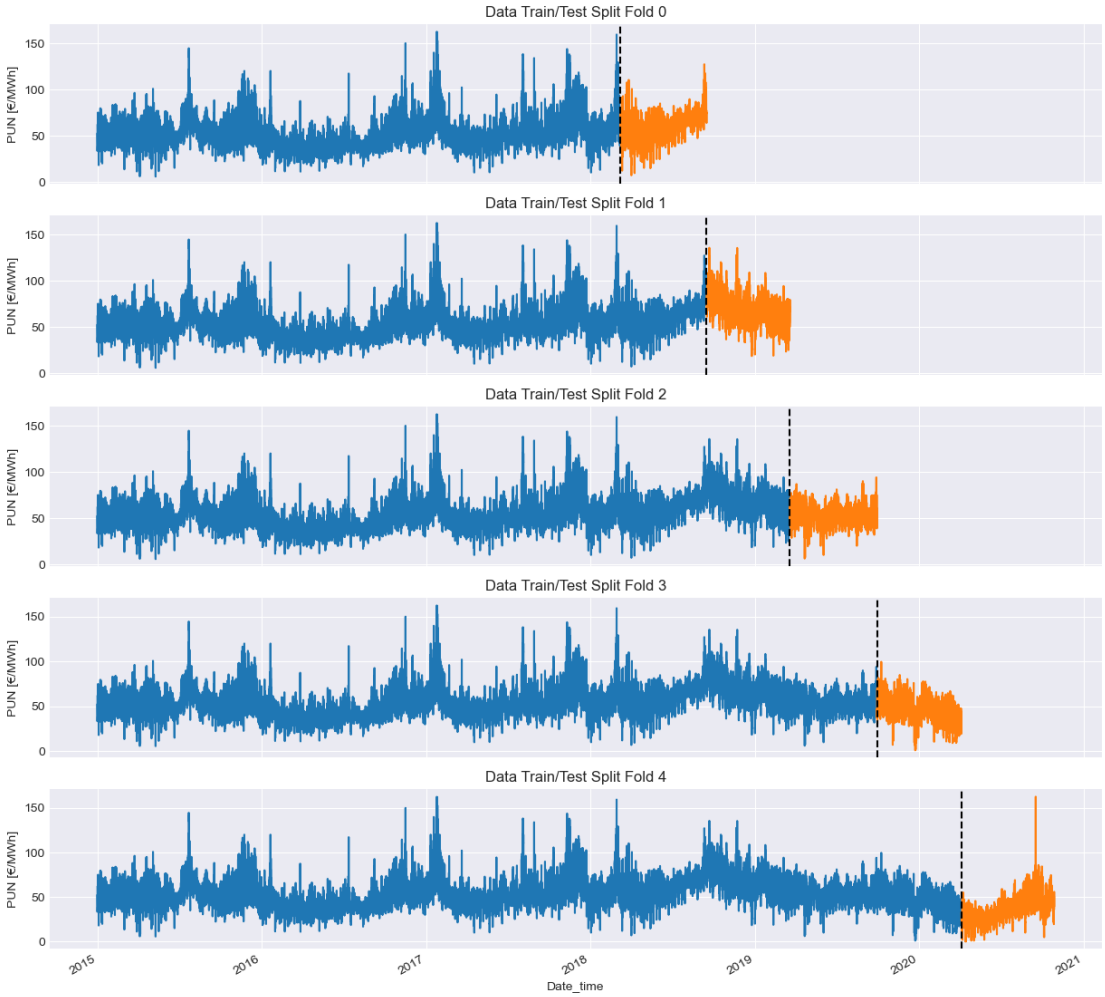

Figura 6.1: Walk-forward Validation

6.3.2 Addestramento del Modello

A seguito della validazione Walk-forward, il modello è stato soggetto ad addestramento e validazione attraverso cinque iterazioni distinte. Per ciascuna di queste iterazioni, è stato calcolato il valore del RMSE al fine di valutare le prestazioni di ogni modello. Questi punteggi sono stati successivamente raccolti in un elenco, consentendo un'analisi dettagliata dell'evoluzione delle prestazioni attraverso i vari fold. Come si puo osservare in Tabella 6.1, nel corso dell'addestramento il punteggio RMSE tende a diminuire, indicando una progressiva riduzione dell'errore commesso dal modello e di conseguenza, un miglioramento delle sue capacità predittive. Infine, è stato calcolato un RMSE medio rappresentativo.

Al fine di realizzare un addestramento efficace, è stata prestata particolare attenzione alla configurazione dei parametri dell'algoritmo, come descritto nello script riportato in Appendice A.4:

- base score: rappresenta il punteggio iniziale per le previsioni, agendo come riferimento all'avvio delle iterazioni. É stato scelto il valore predefinito di 0,5.
- **booster**: identifica il tipo di modello utilizzato per ottimizzare le previsioni. Nel caso specifico, è stato selezionato `gbtree`, che impiega alberi decisionali;
- **n estimators:** indica il numero di alberi decisionali da costruire. Un numero elevato di estimatori può potenzialmente incrementare la precisione del modello, ma è fondamentale stabilire un limite superiore per prevenire fenomeni di overfitting. Il numero di estimatori è stato fissato a 1000;
- **early stopping rounds:** questo parametro, se impostato adeguatamente, permette di interrompere l'addestramento qualora non si verifichi un miglioramento della metrica di valutazione per un numero consecutivo di 50 round, contribuendo così a evitare l'overfitting;
- **objective:** definisce l'obiettivo del modello e influisce direttamente sul calcolo della perdita e sull'ottimizzazione del modello durante l'addestramento. É stata scelta l'opzione `reg:linear`, selezionando un regressore lineare;
- **max depth:** questo parametro controlla la profondità massima di ciascun albero. Un valore maggiore consente al modello di catturare relazioni più complesse tra i dati incrementando, tuttavia, anche il rischio di overfitting. É stato selezionato un valore di 3;
- **learning rate:** conosciuto anche come "tasso di apprendimento", riduce il contributo di ciascun albero all'aggiornamento finale del modello. È importante

che tale valore sia mantenuto basso; sebbene ciò richieda l'utilizzo di un maggior numero di alberi per convergere a una soluzione ottimale, può migliorare la qualità della previsione finale riducendo il rischio di overfitting.

Dopo aver reiterato la fase di addestramento e validazione su tutti i fold, si è ritenuto necessario con un ulteriore addestramento del modello sull'intero set di training al fine di fruttare la totalità dei dati disponibili, con l'obiettivo di ottimizzare ulteriormente le capacità predittive del modello.

Tabella 6.1: Validazione iterativa

RMSE score on Walk-forward validation	
validation-0-rmse:20.79470	validation-1-rmse:27.67132
validation-0-rmse:10.99635	validation-1-rmse:17.51298
validation-0-rmse:8.30803	validation-1-rmse:14.81651
validation-0-rmse:7.50087	validation-1-rmse:14.08672
validation-0-rmse:7.10763	validation-1-rmse:13.68540
validation-0-rmse:6.83541	validation-1-rmse:13.39151
validation-0-rmse:6.62437	validation-1-rmse:13.41239
validation-0-rmse:6.61872	validation-1-rmse:13.4206
validation-0-rmse:21.32922	validation-1-rmse:31.53640
validation-0-rmse:11.36076	validation-1-rmse:17.80166
validation-0-rmse:8.67851	validation-1-rmse:13.02897
validation-0-rmse:7.87703	validation-1-rmse:11.14845
validation-0-rmse:7.47545	validation-1-rmse:9.97053
validation-0-rmse:7.19306	validation-1-rmse:9.43169
validation-0-rmse:6.99721	validation-1-rmse:9.18363
validation-0-rmse:6.82444	validation-1-rmse:9.08815
validation-0-rmse:6.68042	validation-1-rmse:9.05748
validation-0-rmse:6.66281	validation-1-rmse:9.05785
validation-0-rmse:22.05947	validation-1-rmse:15.96
validation-0-rmse:11.66641	validation-1-rmse:8.69485
validation-0-rmse:9.98281	validation-1-rmse:9.45470
validation-0-rmse:22.01108	validation-1-rmse:11.23761
validation-0-rmse:12.43311	validation-1-rmse:11.06330
validation-0-rmse:21.95899	validation-1-rmse:11.26860
validation-0-rmse:12.49574	validation-1-rmse:10.44182
Score across folds 9.7062	

6.3.3 Validazione del Modello

In Tabella 6.2 vengono presentate le performance del modello durante la fase di validazione. Successivamente, sono state calcolate le previsioni sul test set e confrontate con i valori reali (Figura 6.2).

Tabella 6.2: Validazione sul Testing set completo

RMSE score validation set	
validation-0-rmse	52.12029
validation-0-rmse	22.27956
validation-0-rmse	12.74034
validation-0-rmse	10.01755
validation-0-rmse	8.97429
validation-0-rmse	8.47019

6.3.4 Valutazione della Performance

Sono state calcolate,infine le metriche di valutazione, riassunte in Tabella 6.3. In particolare il modello ha un MAE di **6,11**, un R2 di **0,726** e un RMSE di **8,470.**

Tabella 6.3: Metriche di Performance - XGBoost

Valutazione dell'errore	
RMSE	8.470
MAE	6.110
R2	0.726

6.4 Conclusioni

Risulta evidente, dunque, come il modello XGBoost, grazie al suo algoritmo basato su alberi decisionali, ha significativamente migliorato le prestazioni rispetto al modello di regressione "base" inizialmente considerato, portando a una riduzione del valore di RMSE fino a 8,47 e un incremento del R2 di 28,1 punti percentuali. Ciò evidenzia l'efficacia dell'approccio Ensemble nel catturare le complessità e le dinamiche della serie temporale del PUN, risultando in un modello più accurato e attendibile.
In particolare, in Figura 6.2, è possibile notare come, sebbene il modello non sia riuscito a spiegare i picchi massimi e minimi della serie, è stato comunque in grado di spiegarne in parte il trend e la variabilità.

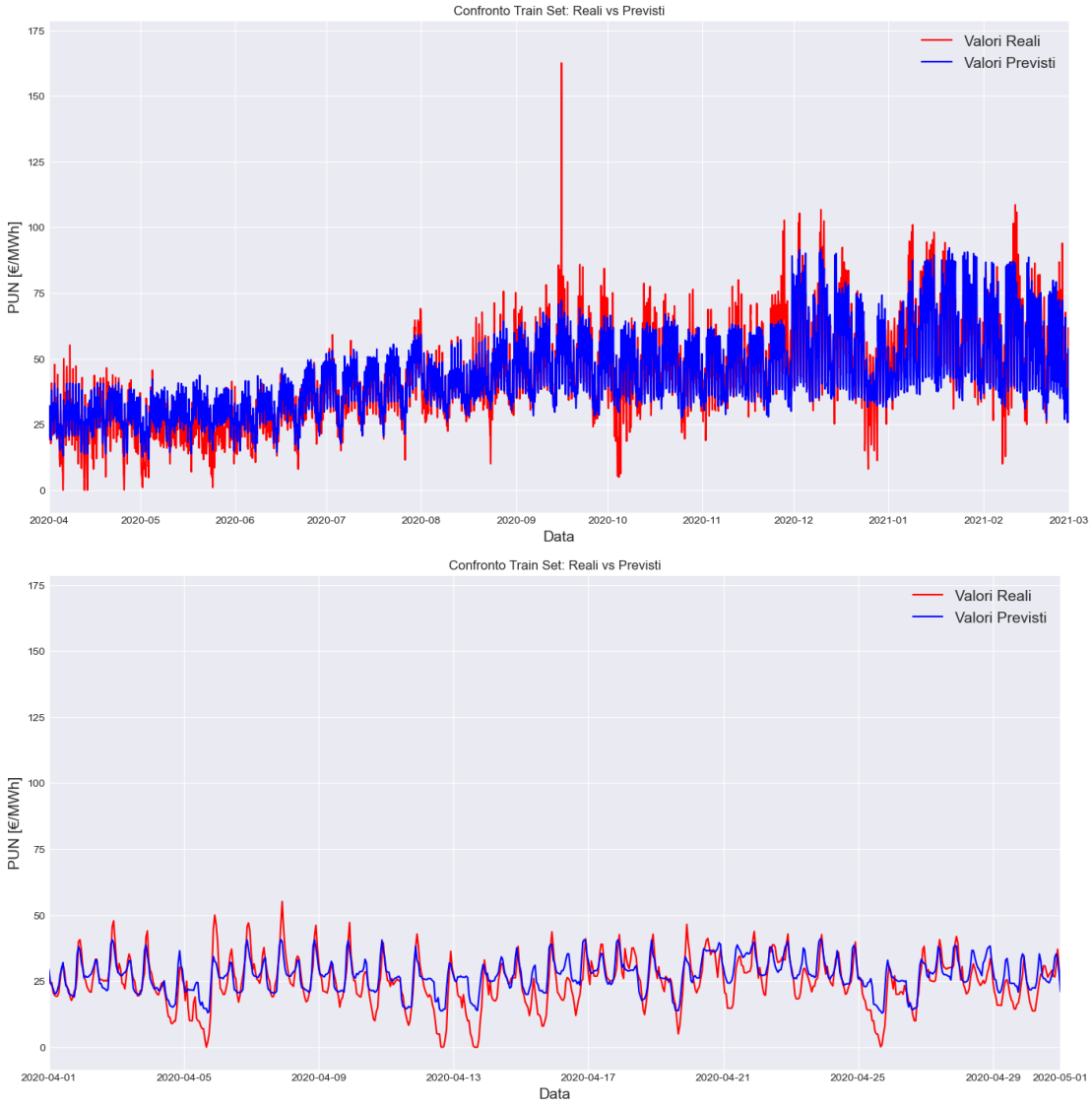

Figura 6.2: Validazione del Modello XGB e confronto tra valori reali e valori predetti

Capitolo 7

Calcolo dei Valori Futuri e Analisi degli Scenari

Questo capitolo conclusivo delibera sul processo di implementazione degli scenari e delle previsioni estese ad un arco temporale decennale, da Marzo 2021 a Marzo 2030.

Dopo aver terminato l'analisi sui modelli oggetto di studio, i risultati vengono riportati in Tabella 7.1 al fine di confrontare le metriche di valutazione di ciascuno di essi.

Ciò che si evince è un progressivo miglioramento delle prestazioni all'aumentare della complessità del modello. In particolare, il modello XGBoost emerge per la sua capacità di spiegare il **72,6%** della varianza della serie temporale, ossia circa il doppio rispetto al modello SARIMA e significativamente superiore rispetto al modello di Regressione Lineare Multipla. Questo rende l'Extreme Gradient Boosting il modello migliore per effettuare la previsione dei valori del Prezzo Unico Nazionale.

Tabella 7.1: Confronto dei risultati

		RMSE	*MAE*	*R2*
SARIMA		37.657	31.721	-
Regressione Lineare Multipla	*A*	18.530	15.921	0.369
	B	16.388	13.923	0.481
Extreme Gradient Boosting		8.470	6.110	0.726

7.1 Definizione degli Scenari

Gli scenari analizzati si basano sulle proiezioni congiunte eseguite da TERNA e SNAM riguardanti l'evoluzione futura del mercato delle commodities energetiche [15]. Tali previsioni si fondano sulle politiche energetiche adottate dall'Unione Europea e dall'Italia per il raggiungimento degli obiettivi di riduzione delle emissioni di gas serra del 55% entro il 2030 e del raggiungimento della Carbon-Neutrality entro il 2050.

Gli scenari delineati sono i seguenti:

- **Scenario 1:** si prevede un incremento del fabbisogno di energia elettrica del 30% entro il 2030. Tale incremento sarebbe principalmente dovuto alla rapida elettrificazione delle utenze domestiche e industriali del paese che potrà essere soddisfatta dalla capacità istallata sempre maggiore di energia rinnovabile al fine di far fronte all'instabilità e volatilità del prezzo del gas naturale;
- **Scenario 2:** si prevede un incremento del prezzo del gas naturale pari a quello verificatosi a valle dell'inizio del conflitto russo-ucraino dello scorso 24 Febbraio 2021. In un contesto simile, che vede l'Italia fortemente dipendente dal gas russo, non si esclude che il prezzo del gas naturale possa aumentare di ben 10 volte rispetto al prezzo di partenza.

Per entrambi gli scenari, si assume che tutte le variabili, ad eccezione di quella specifica di scenario, rimangano costanti. É stato creato, inoltre, un data-set vuoto indicizzato sulle date future (2021-2030), che verrà imputato con i valori previsti del PUN e le assunzioni sulle variabili esogene.

7.1.1 Scenario 1 - Aumento del Fabbisogno Elettrico

Il processo di simulazione dello *Scenario 1* è stato affrontato nel modo seguente:

1. **Domanda di energia elettrica:** una volta analizzata la variazione della domanda nell'ultimo anno del data-set di test (2020-2021) è stato applicato, per semplicità, tale trend a tutti gli anni successivi, assumendo che la domanda di energia elettrica continui a crescere con lo stesso ritmo. Successivamente, al fine di ottenere un aumento complessivo del 30% entro il 2030, è stata sommata una retta lineare di medesima pendenza ai valori presi in esame. A titolo di esempio, il valore minimo della domanda relativo al primo giorno del set di previsione (01/04/2021 00:00:00) è passato da un valore di 23000 MWh ad un valore di 30000 MWh (Figura 7.1);
2. **Prezzo del gas naturale:** per il prezzo del gas naturale, è stato ipotizzato un trend costante simile a quello osservato nell'ultimo anno del data-set di test al fine di mantenere la varianza intrinseca della serie (Figura 7.1);
3. **Produzione da fonti rinnovabili:** per la produzione di energia da fonti rinnovabili è stato adottato un approccio simile a quello del prezzo del gas assumendo che la produzione vari con un trend costante e pari a quello dell'ultimo anno del data-set di test (Figura 7.2).

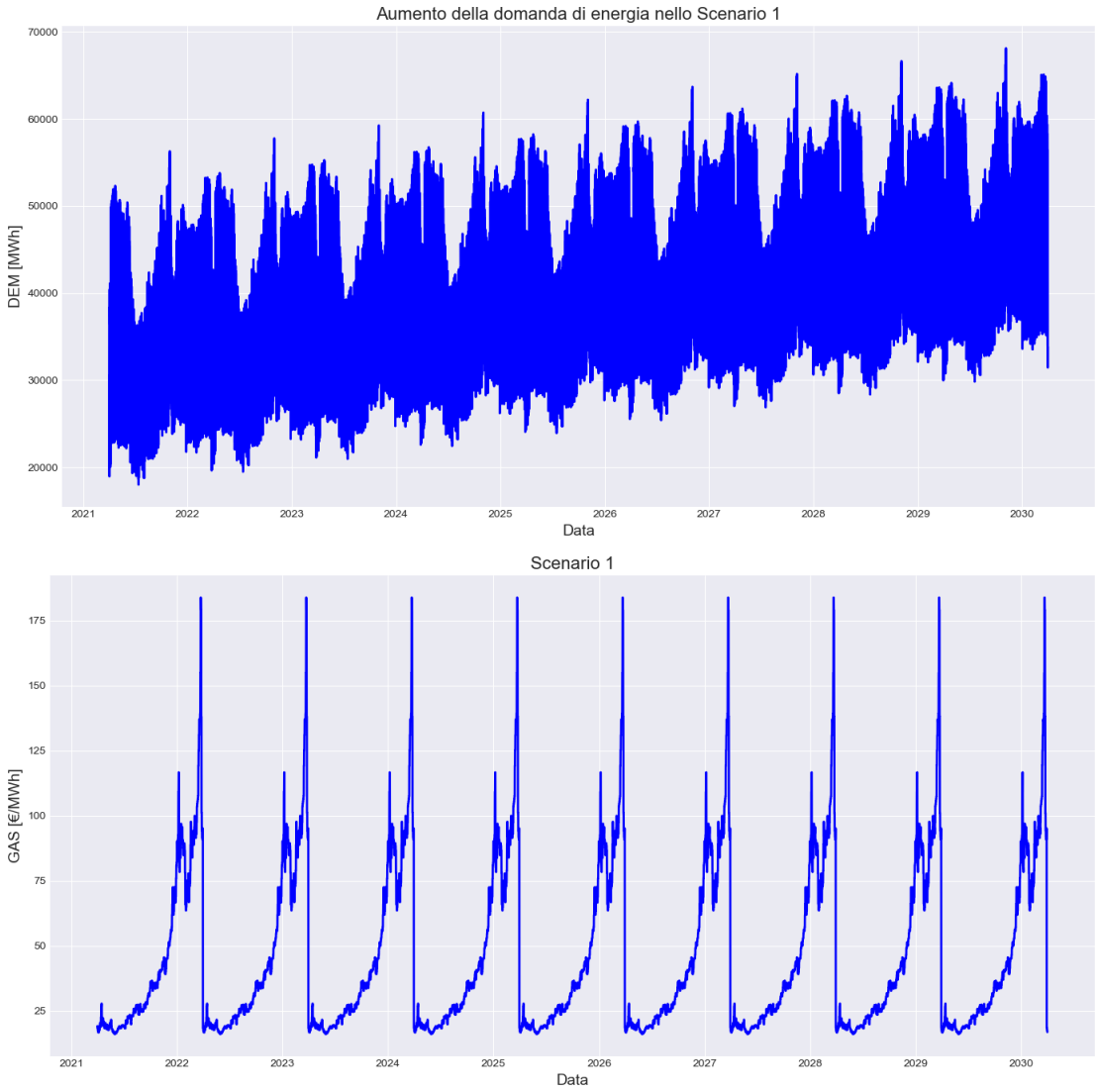

Figura 7.1: Previsione del fabbisogno elettrico (sopra) e del gas naturale (sotto) - Scenario 1

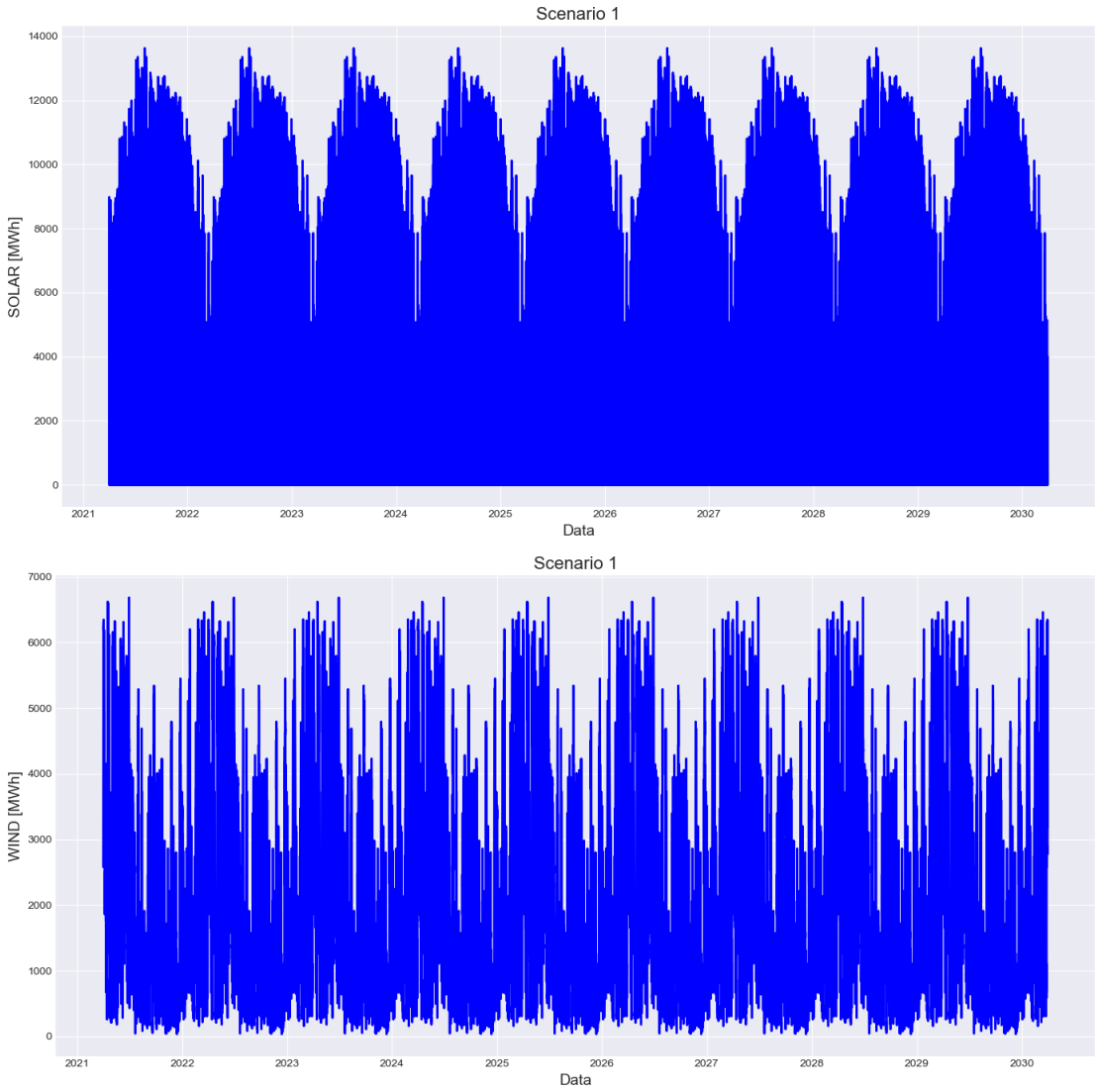

Figura 7.2: Previsione dell'energia generata da fotovoltaico (sopra) ed eolico (sotto) - Scenario 1

Calcolo delle Previsioni e Analisi dei Risultati

Le previsioni sono state generate utilizzando la funzione `reg.predict`, confermando le teorie economiche secondo le quali l'aumento della domanda di energia, a parità di offerta, tende a far aumentare il prezzo. I risultati, mostrano infatti che il PUN tenderebbe ad aumentare del 30-40% nello scenario considerato (Figura 7.3).

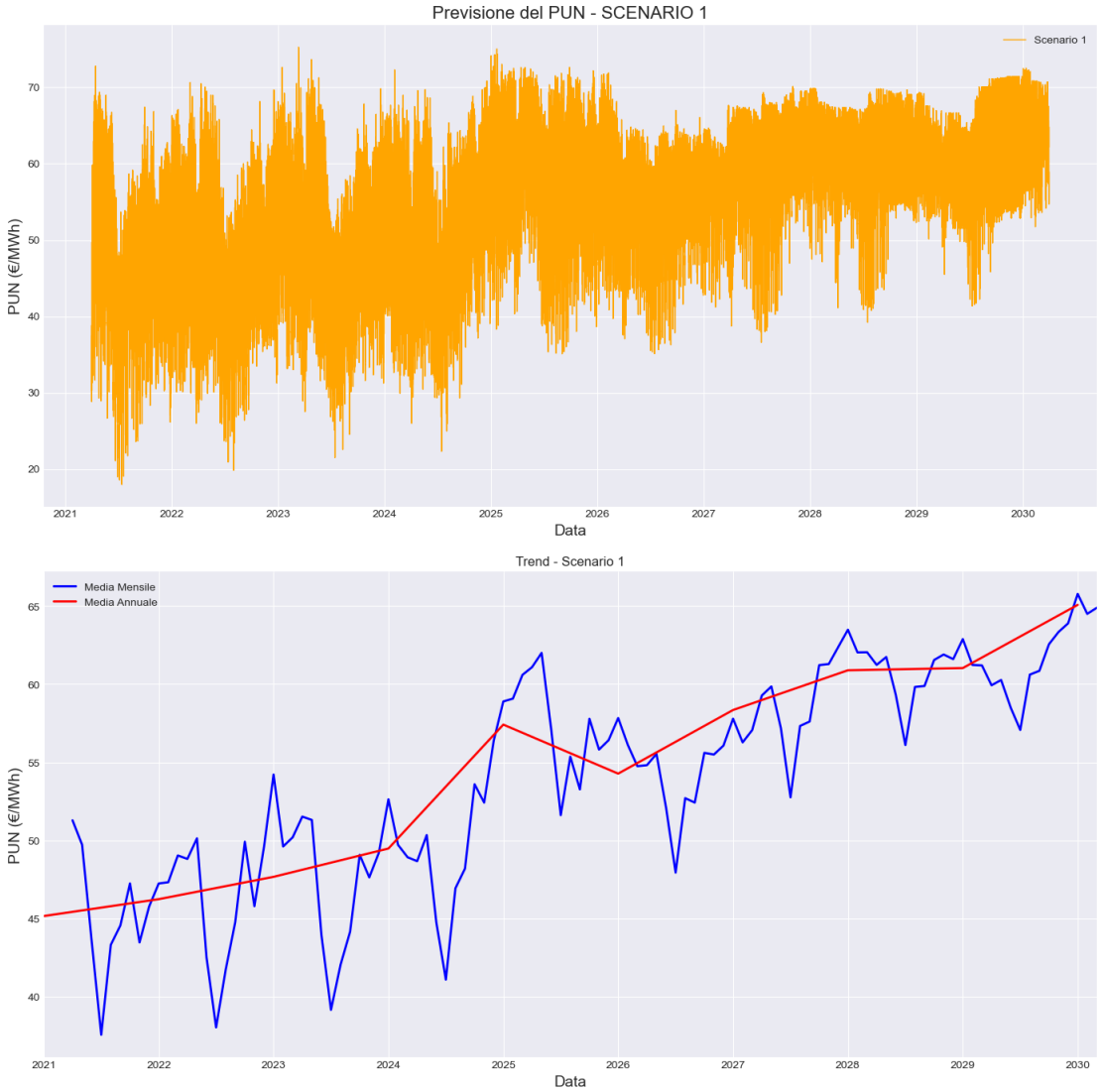

Figura 7.3: Previsione del PUN - Scenario 1

7.1.2 Scenario 2 - Aumento del Prezzo MGP del Gas Naturale

Per simulare l'aumento vertiginoso del prezzo del gas naturale (*Scenario 2*) è stato adottato un approccio graduale simile a quello utilizzato per lo Scenario 1, ossia:

1. **Prezzo del gas naturale:** una volta analizzata la variazione del prezzo del gas naturale nell'ultimo anno del data-set di test (2020-2021) per semplicità è stato applicato tale trend a tutti gli anni successivi, assumendo che esso continui a crescere con lo stesso ritmo. Successivamente, è stata sommata una retta lineare con una pendenza tale al fine di ottenere un aumento complessivo del 1000% (10 volte il prezzo di partenza) entro il 2030. A titolo di esempio, il valore minimo del prezzo del gas naturale relativo al primo giorno del set di previsione (01/04/2021 00:00:00) è passato da un valore di 30 €/MWh ad un valore di 350 €/MWh (Figura 7.4).
2. **Domanda di energia:** i valori della domanda di energia sono stati resettati ai valori originali seppur considerando un trend costante simile a quello osservato nell'ultimo anno del data-set di test (Figura 7.4).
3. **Produzione da fonti rinnovabili:** per la produzione di energia da fonti rinnovabili, i valori sono assunti pari a quanto visto per lo Scenario 1 (Figura 7.2).

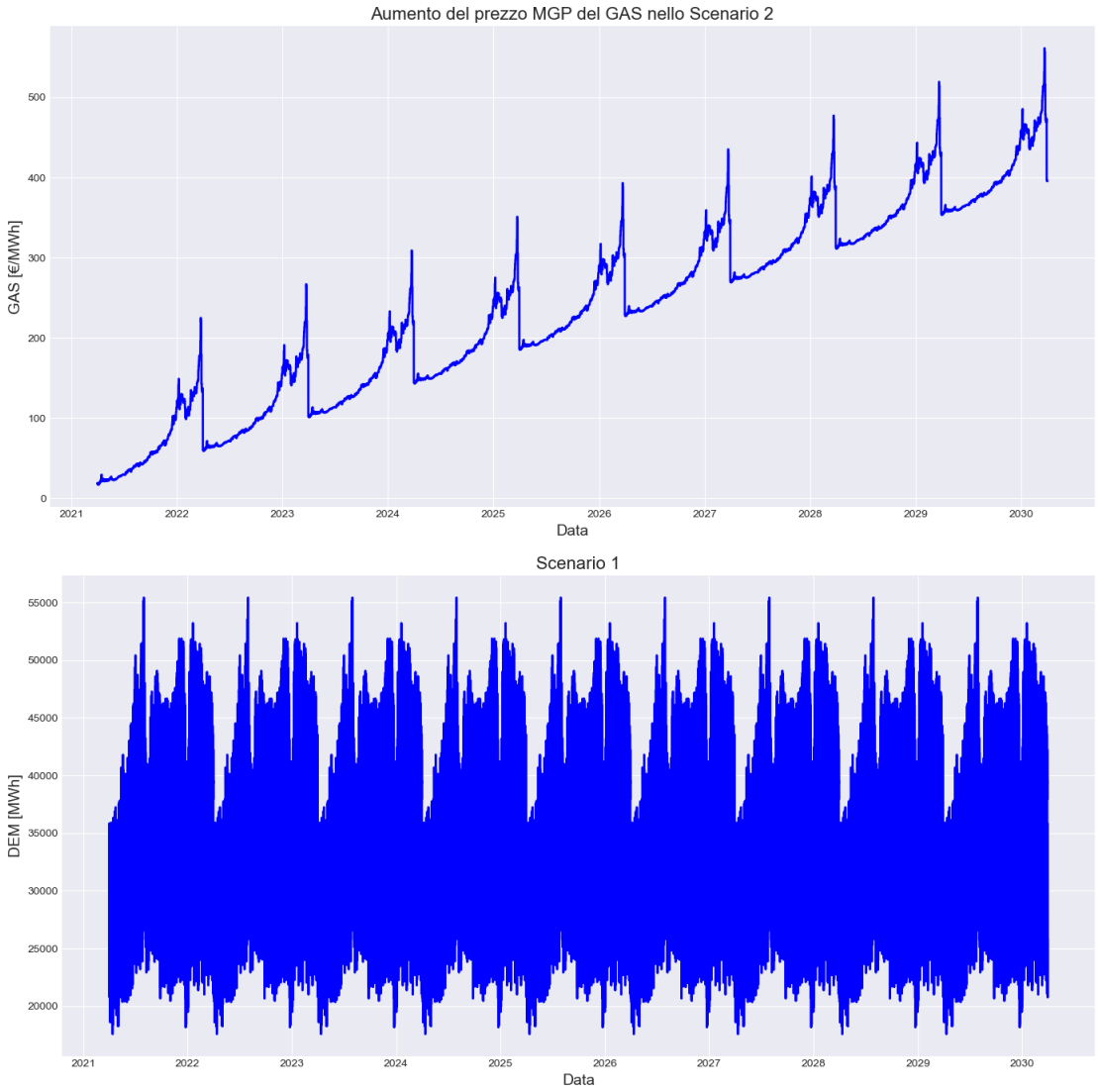

Figura 7.4: Previsione del prezzo MGP del gas naturale (sopra) e del fabbisogno elettrico (sotto) - Scenario 2

Calcolo delle Previsioni e Analisi dei Risultati

Le previsioni, generate tramite la funzione `reg.predict`, confermano quanto è stato osservato nell'ultimo biennio con il caro energia. I risultati, mostrano infatti che il PUN tenderebbe aumentare del ben 60% in uno scenario di aumento del prezzo del gas naturale (Figura 7.5).

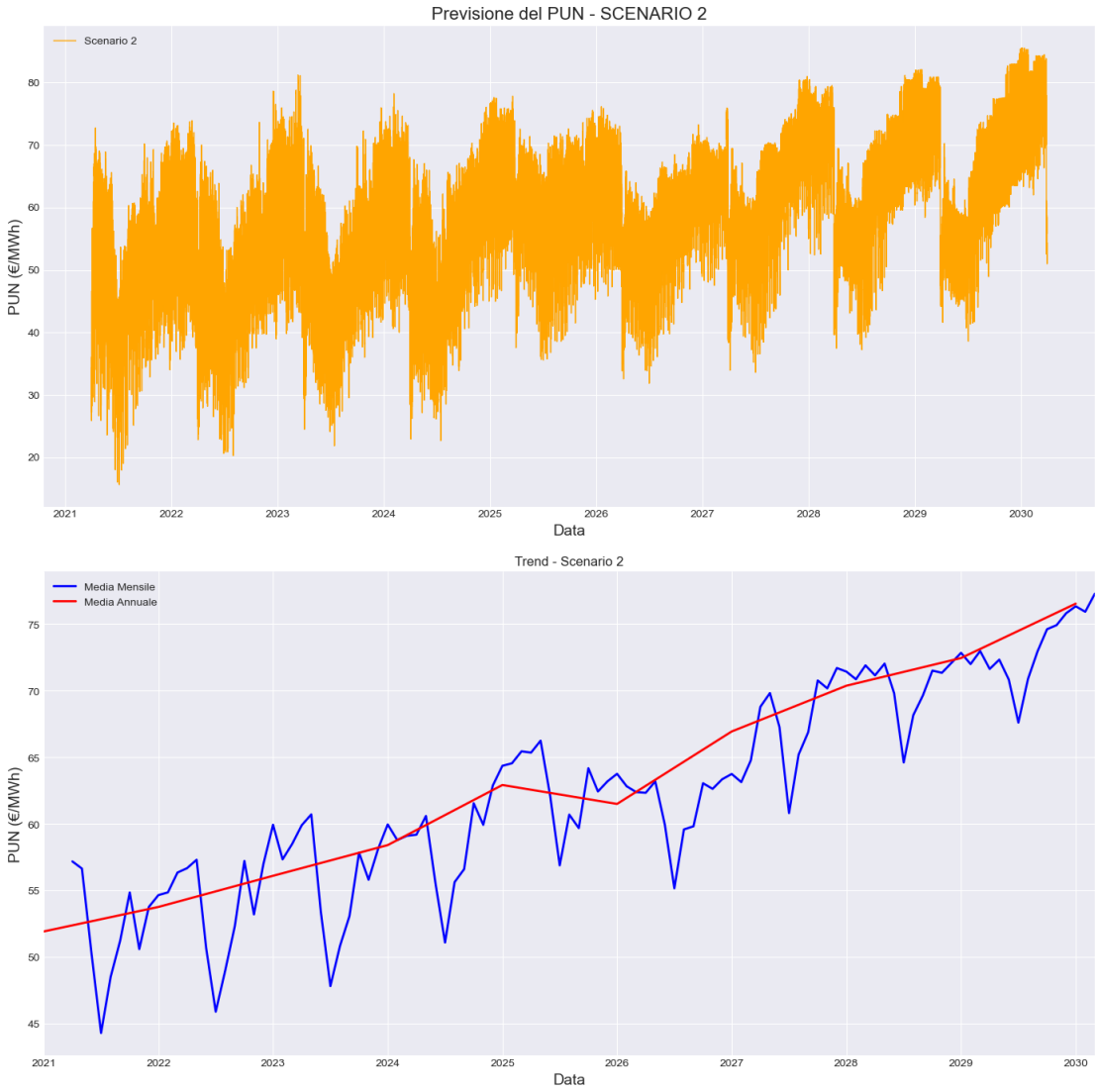

Figura 7.5: Previsione del PUN - Scenario 2

7.2 Considerazioni Finali

Sula base delle analisi effettuate finora emergono due scenari critici.

Nel primo scenario, ipotizzando un aumento del fabbisogno energetico del 30% entro il 2030, il PUN tenderebbe ad aumentare del 40%; nel secondo scenario, un aumento del prezzo del gas naturale impatterebbe in modo ancora più drammatico, con un potenziale aumento del PUN del 60% (Figura 7.6).
In entrambi gli scenari, dunque, le conseguenze per famiglie e imprese sarebbero significative, con un impatto negativo sul potere d'acquisto e sulla competitività del sistema produttivo italiano.

Appare evidente, dunque, come l'Italia si trovi di fronte ad una congiuntura cruciale nel settore energetico, caratterizzata da una serie di sfide imminenti. Da un lato, si evidenzia una crescente domanda di energia elettrica, attribuibile in parte all'aumento demografico e alla progressiva elettrificazione del territorio nazionale; dall'altro, la marcata dipendenza dal gas naturale, esacerbata dalle recenti tensioni geopolitiche, espone il sistema energetico nazionale a vulnerabilità derivanti dalle fluttuazioni del mercato internazionale.

Di fronte a queste sfide, la risposta appare inequivocabile: diversificare le fonti di approvvigionamento energetico e accelerare la transizione verso un sistema energetico più resiliente e sostenibile.
La diversificazione implica un impegno massiccio nelle fonti rinnovabili, quali l'eolico, il fotovoltaico, l'idroelettrico, il geotermico e le biomasse. Parallelamente, si auspica la stipula di accordi con nuovi fornitori di gas al fine di ridurre la dipendenza da un unico paese oltre a investire nello sviluppo di tecnologie volte all'utilizzo di gas alternativi, come il biometano e l'idrogeno.
L'accelerazione della transizione energetica richiede, invece, la promozione dell'elettrificazione per il riscaldamento e gli usi domestici, al fine di eliminare l'uso del gas naturale perlomeno in questi settori. È altresì necessario sviluppare infrastrutture di ricarica per veicoli elettrici, nonché investire in ricerca e sviluppo di tecnologie ecologiche per il trasporto di merci pesanti.

Si tratta di un percorso non privo di complessità, ma imprescindibile per garantire la sicurezza energetica, che si configura non solo come una questione economica, ma anche come una priorità di sicurezza nazionale e di tutela ambientale. Un sistema energetico diversificato e resiliente costituisce un pilastro fondamentale per la prosperità economica e per la salvaguardia del patrimonio ambientale nazionale.

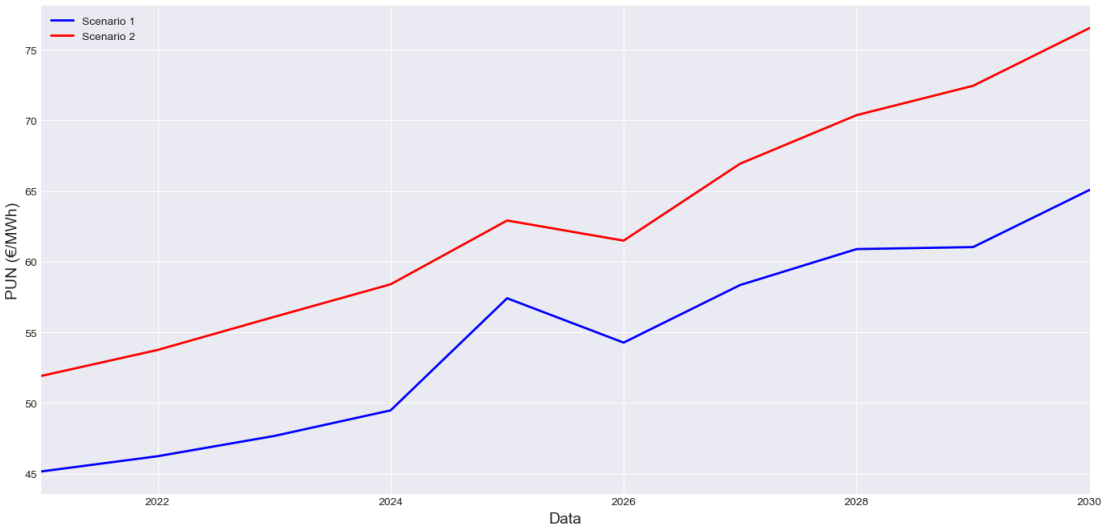

Figura 7.6: Confronto degli scenari

Capitolo 8

Conclusioni e Sviluppi Futuri

L'obiettivo principale dell'elaborato consisteva nell'esplorare alcune tecniche previsionali studiate in letteratura, dopo una preliminare introduzione e presentazione del contesto normativo, nonché la definizione della struttura del mercato elettrico al fine di di condurre un'analisi di previsione del Prezzo Unico Nazionale (PUN) in due scenari definiti come potenzialmente possibili.

In particolare, le analisi condotte hanno evidenziato un miglioramento delle performance dei modelli all'aumentare del numero di features considerate. Nel contesto dell'analisi previsionale, infatti, il modello Extreme Gradient Boosting si è distinto per un basso valore di RMSE e un R2 significativamente superiore rispetto agli altri modelli presi in considerazione.
L'inclusione di queste tecniche nell'ottica di una programmazione della produzione energetica, soprattutto per impianti termoelettrici o idroelettrici ad accumulo, può rivestire un ruolo fondamentale. La capacità di prevedere l'andamento del prezzo dell'energia elettrica consente, inoltre, di ottimizzare la pianificazione e la diversificazione della produzione nell'arco della giornata, contribuendo a massimizzare il ritorno economico derivante dalla vendita dell'energia prodotta.

Tuttavia, è importante sottolineare che la stima del prezzo dell'energia, immune da qualsiasi errore, rimane un compito estremamente complesso. Come discusso nel capitolo precedente, il prezzo dell'energia è soggetto a una considerevole volatilità ed è influenzato da fattori spesso imprevedibili. Cio nonstante, i modelli sviluppati, pur essendo affetti da errori, possono fungere come punto di partenza per comprendere l'andamento generale di questa grandezza.

È possibile, infine, delineare diverse direzioni di sviluppo future. Tra queste,

risulterebbe interessante esaminare come l'aumento della produzione di energia da fonti rinnovabili non programmabili, quali fotovoltaico ed eolico, potrebbe influenzare la previsione, attendendosi , ragionevolmente, che un loro aumento possa contribuire a ridurre il Prezzo Unico del mercato elettrico italiano.

Appendice A

Codici di programmazione

A.1 Pre-processing

```
LIBRERIE
import pandas as pd
import numpy as np
import os
```

```
ESTRAPOLAZIONE E CARICAMENTO DEI DATI
directory
PUN_data = os.listdir(directory)
PUN_df = {}

for file in PUN_data:
    path = os.path.join(directory, file)
    PUN_df[file] = pd.read_excel(path, engine='openpyxl')
    print(PUN_df.keys())

print(f"Numero di file caricati: {len(PUN_df)}")

PUN_all_data = []
for file in PUN_data:
    path = os.path.join(directory, file)
    try:
        df = pd.read_excel(path, sheet_name="Prezzi-Prices", engine='
    openpyxl' if file.endswith('.xlsx') else 'xlrd')
        PUN_all_data.append(df)
```

```
19          print(f"File {file} e foglio 'Prezzi-Prices' caricato con
    successo!")
20      except Exception as e:
21          print(f"Errore durante la lettura del file {file} e del
    foglio 'Prezzi-Prices': {e}")
22
23 PUN_15_21 = pd.concat(PUN_all_data, ignore_index=True)
24 PUN_15_21 = PUN_15_21[['    Data/Date\n(YYYYMMDD)', 'Ora\n/Hour', 'PUN
    ']]
25
26 CONVERSIONE E INDICIZZAZIONE DELLA COLONNA DATE_TIME
27 PUN_15_21['    Data/Date\n(YYYYMMDD)'] = pd.to_datetime(PUN_ALL_15_21[
    '    Data/Date\n(YYYYMMDD)'], format='%Y%m%d').dt.strftime('%d-%m-%
    Y')
28 PUN_15_21['Ora\n/Hour'] = PUN_15_21['Ora\n/Hour'].apply(lambda x: f"{
    x:02d}:00")
29 PUN_15_21['Date_time'] = PUN_15_21['    Data/Date\n(YYYYMMDD)'] + ' '
    + PUN_15_21['Ora\n/Hour']
30 PUN_15_21.drop(columns=['    Data/Date\n(YYYYMMDD)', 'Ora\n/Hour'],
    inplace=True)
31
32 PUN_15_21.set_index('Date_time', inplace=True)
33
34 RAPPRESENTAZIONE DELLE VARIABILI
35
36 plt.figure(figsize=(14, 7))
37 plt.style.use('seaborn-darkgrid')
38 plt.plot(ds_15_21.index, ds_15_21['PUN'], label='PUN', color='orange'
    , linewidth=2)
39 plt.title('Andamento del PUN  nel tempo', fontsize=16)
40 plt.xlabel('Data', fontsize=14)
41 plt.ylabel('PUN ', fontsize=14)
42 plt.legend()
43 plt.grid(True)
44 plt.tight_layout()
45 plt.show()
46
47 plt.figure(figsize=(14, 7))
48 plt.style.use('seaborn-darkgrid')
49 plt.plot(ds_15_21.index, ds_15_21['DEM'], label='DEM', color='blue',
    linewidth=2)
50 plt.title('Andamento del fabbisogno energetico (MWh) nel tempo',
    fontsize=16)
51 plt.xlabel('Data', fontsize=14)
52 plt.ylabel('DEM (MWh)', fontsize=14)
53 plt.legend()
54 plt.grid(True)
55 plt.tight_layout()
56 plt.show()
```

```
57
58 plt.figure(figsize=(14, 7))
59 plt.style.use('seaborn-darkgrid')
60 plt.plot(ds_15_21.index, ds_15_21['GAS'], label='GAS', color='green',
       linewidth=2)
61 plt.title('Andamento del Prezzo del GAS  nel tempo', fontsize=16)
62 plt.xlabel('Data', fontsize=14)
63 plt.ylabel('GAS ', fontsize=14)
64 plt.legend()
65 plt.grid(True)
66 plt.tight_layout()
67 plt.show()
68
69 plt.figure(figsize=(14, 7))
70 plt.style.use('seaborn-darkgrid')
71 plt.plot(ds_15_21.index, ds_15_21['SOLAR'], label='PUN', color='red',
       linewidth=2)
72 plt.title('Andamento dell energia prodotta da fotovoltaico (MWh) nel
     tempo', fontsize=16)
73 plt.xlabel('Data', fontsize=14)
74 plt.ylabel('SOLAR (MWh)', fontsize=14)
75 plt.legend()
76 plt.grid(True)
77 plt.tight_layout()
78 plt.show()
79
80 plt.figure(figsize=(14, 7))
81 plt.style.use('seaborn-darkgrid')
82 plt.plot(ds_15_21.index, ds_15_21['WIND'], label='PUN', color='violet
     ', linewidth=2)
83 plt.title('Andamento dell energia prodotta da eolico (MWh) nel tempo'
     , fontsize=16)
84 plt.xlabel('Data', fontsize=14)
85 plt.ylabel('WIND (MWh)', fontsize=14)
86 plt.legend()
87 plt.grid(True)
88 plt.tight_layout()
89 plt.show()
90
91 BOXPLOT
92 ### PUN
93 fig, ax = plt.subplots(figsize=(14,7))
94 sns.boxplot(data=ds_15_21, x='hour', y='PUN')
95 plt.xlabel('Ora')
96 plt.ylabel('PUN ')
97 ax.set_title('BoxPlot PUN Orario')
98 plt.show()
99
100 fig, ax = plt.subplots(figsize=(14,7))
```

```
sns.boxplot(data=ds_15_21, x='month', y='PUN', palette='Blues')
plt.xlabel('Mesi')
plt.ylabel('PUN ')
ax.set_title('BoxPlot PUN Mensile')
plt.show()

### DEM
fig, ax = plt.subplots(figsize=(14,7))
sns.boxplot(data=ds_15_21, x='hour', y='DEM')
plt.xlabel('Ora')
plt.ylabel('DEM [MWh]')
ax.set_title('BoxPlot DEM Orario')
plt.show()

fig, ax = plt.subplots(figsize=(14,7))
sns.boxplot(data=ds_15_21, x='month', y='DEM', palette='Blues')
plt.xlabel('Mesi')
plt.ylabel('DEM [MWh]')
ax.set_title('BoxPlot DEM Mensile')
plt.show()

### GAS
fig, ax = plt.subplots(figsize=(14,7))
sns.boxplot(data=ds_15_21, x='month', y='GAS', palette='Blues')
plt.xlabel('Mesi')
plt.ylabel('GAS ')
ax.set_title('BoxPlot GAS Mensile')
plt.show()

### SOLAR
fig, ax = plt.subplots(figsize=(14,7))
sns.boxplot(data=ds_15_21, x='hour', y='SOLAR')
plt.xlabel('Ora')
plt.ylabel('SOLAR [MWh]')
ax.set_title('BoxPlot SOLAR Orario')
plt.show()

fig, ax = plt.subplots(figsize=(14,7))
sns.boxplot(data=ds_15_21, x='month', y='SOLAR', palette='Blues')
plt.xlabel('Mesi')
plt.ylabel('SOLAR [MWh]')
ax.set_title('BoxPlot SOLAR Mensile')
plt.show()

### WIND
fig, ax = plt.subplots(figsize=(14,7))
sns.boxplot(data=ds_15_21, x='hour', y='WIND')
plt.xlabel('Ora')
plt.ylabel('WIND [MWh]')
```

```
ax.set_title('BoxPlot WIND Orario')
plt.show()

fig, ax = plt.subplots(figsize=(14,7))
sns.boxplot(data=ds_15_21, x='month', y='WIND', palette='Blues')
plt.xlabel('Mesi')
plt.ylabel('WIND [MWh]')
ax.set_title('BoxPlot WIND Mensile')
plt.show()

TRAINING E TEST SET
train = ds_15_21['PUN'].loc['2015-01-01 00:00:00':'2020-03-31
    23:00:00']
test = ds_15_21['PUN'].loc['2020-04-01 00:00:00':'2021-03-31 23:00:00
    ']

plt.figure(figsize=(14, 7))
plt.style.use('seaborn-darkgrid')
plt.plot(train.index, train, label='Training Set', color='blue',
    linewidth=2)
plt.plot(test.index, test, label='Test Set', color='red', linewidth
    =2)
plt.title('Suddivisione in Train set e Test set', fontsize=16)
plt.xlabel('Data e ora', fontsize=14)
plt.ylabel('PUN ', fontsize=14)
plt.axvline(pd.Timestamp('2020-03-31'), color='black', ls='--')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

A.2 Seasonal Autoregressive Integrated Moving Average

```
LIBRERIE
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import warnings
import statsmodels.api as sm
import itertools
import os
import pandas as pd
import statsmodels.api as sm
import numpy as np
import seaborn as sns

from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.ar_model import AutoReg
from statsmodels.tsa.stattools import acf, pacf
from statsmodels.tsa.statespace.sarimax import SARIMAX
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from plotly import graph_objs as go
from matplotlib.pylab import rcParams
warnings.filterwarnings("ignore")

plt.rcParams['figure.figsize'] = 14, 7
```

```
TEST ADF
def adf_test(series, title=''):
    print(f'Augmented Dickey-Fuller Test: {title}')
    result = adfuller(series, autolag='AIC')  # AIC selezionerà il
    lag ottimale
    labels = ['ADF test statistic', 'p-value', '# lags used', '#
    observations']
    out = pd.Series(result[0:4], index=labels)

    for key, val in result[4].items():
        out[f'critical value ({key})'] = val

    print(out.to_string())  # to_string() rimuove l'indicazione dtype

    if result[1] <= 0.05:
```

```
14          print("Strong evidence against the null hypothesis")
15          print("Reject the null hypothesis")
16          print("Data has no unit root and is stationary")
17      else:
18          print("Weak evidence against the null hypothesis")
19          print("Fail to reject the null hypothesis")
20          print("Data has a unit root and is non-stationary")
21      print("\n")
22
23  adf_test(PUN_15_21['PUN'], 'PUN')
24
25  ACF E PACF PLOTS
26  fig, ax = plt.subplots(2,1)
27  plt.xticks(range(0, 49, 12))  # Etichetta ogni ora sull'asse x
28  fig = sm.graphics.tsa.plot_acf(PUN_15_21['PUN'].dropna(), lags=50, ax
        =ax[0])
29  plt.xlabel('Lags', fontsize=14)
30
31  plt.xticks(range(0, 49, 12))  # Etichetta ogni ora sull'asse x
32  fig = sm.graphics.tsa.plot_pacf(PUN_15_21['PUN'].dropna(), lags=50,
        ax=ax[1])
33  plt.xlabel('Lags', fontsize=14)
34  plt.grid(True)
35  plt.show()
36
37  TRAIN E TEST SET
38  train = PUN_15_21['PUN'].loc['2015-01-01 00:00:00':'2020-03-31
        23:00:00']
39  test = PUN_15_21['PUN'].loc['2020-04-01 00:00:00':'2021-03-31
        23:00:00']
40
41  fig, ax = plt.subplots(figsize=(15, 5))
42  train.plot(ax=ax, label='Training Set', title='Data Train/Test Split'
        )
43  test.plot(ax=ax, label='Test Set')
44  ax.axvline('2020-03-31', color='black', ls='--')
45  ax.legend(['Training Set', 'Test Set'])
46  plt.show()
47
48  SCELTA DEI PARAMETRI
49  model = auto_arima(PUN_15_21['PUN'], seasonal=True, m=24,  # m è la
        periodicità stagionale, impostala in base al tuo dataset
50                     trace=True, error_action='ignore',
        suppress_warnings=True)
51
52   ANALISI DEI RESIDUI
53  residui = results.resid
54
55  # Grafico dei residui
```

```
plt.figure(figsize=(10, 4))
plt.plot(residui)
plt.title('Residui del Modello')
plt.axhline(0, linestyle='--', color='red')
plt.show()

# Grafico ACF dei residui
sm.graphics.tsa.plot_acf(residui, lags=40)
plt.show()

# Grafico PACF dei residui
sm.graphics.tsa.plot_pacf(residui, lags=40)
plt.show()

# Grafico Q-Q
plt.figure(figsize=(10, 4))
sm.qqplot(residui, line='s', ax=plt.gca())
plt.title('Grafico Q-Q dei Residui')
plt.show()

from scipy import stats
w, p_value = stats.shapiro(residui.dropna()[:5000])  # limitato a
    5000 campioni per prestazioni
print(f'Test di Shapiro-Wilk: statistic = {w}, p-value = {p_value}')

CALCOLO PREVISIONI SUL TEST SET
num_periods = len(test)
predictions = results.get_forecast(steps=num_periods)
predicted_mean = predictions.predicted_mean
prediction_index = pd.date_range(start='2020-04-01 00:00:00', periods
    =num_periods, freq='H')

plt.figure(figsize=(15, 5))
plt.plot(test.index, test, label='Valori Reali', color='blue',
    linewidth=0.5)
plt.plot(prediction_index, predicted_mean, label='Previsti', color='
    red', linewidth=0.5)
plt.legend()
plt.show()

VALUTAZIONE DEL MODELLO
from math import sqrt
test.mean()
rmse=sqrt(mean_squared_error(predicted_mean,test))
print(f"RMSE: {rmse}")

mae_test = mean_absolute_error(test, predicted_mean )
print(f"MAE: {mae}")
```

A.3 Regressione Lineare Multipla

```
DECOMPOSIZIONE
from statsmodels.tsa.seasonal import seasonal_decompose
from scipy.stats import pearsonr

decompositions = {}
columns_to_analyze = ['PUN', 'DEM', 'SOLAR', 'WIND', 'GAS']

for column in columns_to_analyze:
    decompositions[column] = seasonal_decompose(PUN_ALL_15_21[column], model='additive', period=24)
correlation_matrix = PUN_ALL_15_21[columns_to_analyze].corr(method='pearson')

decomposition_results = {}
for column, decomposition in decompositions.items():
    decomposition_results[column] = {
        'trend': decomposition.trend.dropna().head(),  # Display only the first few values of the trend
        'seasonal': decomposition.seasonal.head(),     # Display only the first few values of the seasonal component
        'residual': decomposition.resid.dropna().head() # Display only the first few values of the residuals
    }
decomposition_results, correlation_matrix

CREAZIONE DUMMIES TEMPORALI
def create_features(PUN_ALL_15_21):
    """
    Create time series features based on time series index.
    """
    PUN_ALL_15_21 = PUN_ALL_15_21.copy()
    PUN_ALL_15_21['hour'] = PUN_ALL_15_21.index.hour
    PUN_ALL_15_21['dayofweek'] = PUN_ALL_15_21.index.dayofweek
    PUN_ALL_15_21['quarter'] = PUN_ALL_15_21.index.quarter
    PUN_ALL_15_21['month'] = PUN_ALL_15_21.index.month
    PUN_ALL_15_21['year'] = PUN_ALL_15_21.index.year
    PUN_ALL_15_21['dayofyear'] = PUN_ALL_15_21.index.dayofyear
    PUN_ALL_15_21['dayofmonth'] = PUN_ALL_15_21.index.day
    PUN_ALL_15_21['weekofyear'] = PUN_ALL_15_21.index.isocalendar().week
    return PUN_ALL_15_21

PUN_ALL_15_21 = create_features(PUN_ALL_15_21)
PUN_ALL_15_21
```

```
39
40 TRAIN E TEST SET
41 train = PUN_ALL_15_21['PUN'].loc['2015-01-01 00:00:00':'2020-03-31
       23:00:00']
42 test = PUN_ALL_15_21['PUN'].loc['2020-04-01 00:00:00':'2021-03-31
       23:00:00']
43
44 fig, ax = plt.subplots(figsize=(15, 5))
45 train.plot(ax=ax, label='Training Set', title='Data Train/Test Split'
       )
46 test.plot(ax=ax, label='Test Set')
47 ax.axvline('2020-03-31', color='black', ls='--')
48 ax.legend(['Training Set', 'Test Set'])
49 plt.show()
50
51 ##DEM e GAS
52 X = sm.add_constant(PUN_ALL_15_21_DG[['DEM', 'GAS','dayofyear', 'hour
       ', 'dayofweek', 'quarter', 'month', 'year']])
53 X_train = X[['DEM', 'GAS','dayofyear', 'hour', 'dayofweek', 'quarter'
       , 'month', 'year']].loc['2015-01-01 00:00:00':'2020-03-31 23:00:00
       ']
54 X_train_const = sm.add_constant(X_train)  # Aggiungi una costante al
       modello
55
56 X_test = X[['DEM', 'GAS','dayofyear', 'hour', 'dayofweek', 'quarter',
        'month', 'year']].loc['2020-04-01 00:00:00':'2021-03-31 23:00:00'
       ]
57 X_test_const = sm.add_constant(X_test)  # Aggiungi una costante al
       modello
58
59 ##DEM, GAS, SOLAR, WIND
60 X_all = sm.add_constant(PUN_ALL_15_21[['DEM', 'GAS','SOLAR', 'WIND','
       dayofyear', 'hour', 'dayofweek', 'quarter', 'month', 'year']])
61 X_all_train = X_all[['DEM', 'GAS','SOLAR', 'WIND','dayofyear', 'hour'
       , 'dayofweek', 'quarter', 'month', 'year']].loc['2015-01-01
       00:00:00':'2020-03-31 23:00:00']
62 X_all_train_const = sm.add_constant(X_all_train)  # Aggiungi una
       costante al modello
63
64 X_all_test = X_all[['DEM', 'GAS','SOLAR', 'WIND','dayofyear', 'hour',
        'dayofweek', 'quarter', 'month', 'year']].loc['2020-04-01
       00:00:00':'2021-03-31 23:00:00']
65 X_all_test_const = sm.add_constant(X_all_test)  # Aggiungi una
       costante al modello
66
67 TEST ADF
68 from statsmodels.tsa.stattools import adfuller
69
70 def adf_test(series, title=''):
```

```
71      print(f'Augmented Dickey-Fuller Test: {title}')
72      result = adfuller(series, autolag='AIC')  # AIC selezionerà il
        lag ottimale
73      labels = ['ADF test statistic', 'p-value', '# lags used', '#
        observations']
74      out = pd.Series(result[0:4], index=labels)
75
76      for key, val in result[4].items():
77          out[f'critical value ({key})'] = val
78
79      print(out.to_string())  # to_string() rimuove l'indicazione dtype
80
81      if result[1] <= 0.05:
82          print("Strong evidence against the null hypothesis")
83          print("Reject the null hypothesis")
84          print("Data has no unit root and is stationary")
85      else:
86          print("Weak evidence against the null hypothesis")
87          print("Fail to reject the null hypothesis")
88          print("Data has a unit root and is non-stationary")
89      print("\n")
90
91  adf_test(PUN_ALL_15_21_DG['PUN'], 'PUN')
92  adf_test(PUN_ALL_15_21_DG['DEM'], 'DEM')
93  adf_test(PUN_ALL_15_21_DG['GAS'], 'GAS')
94  adf_test(PUN_ALL_15_21_DG['SOLAR'], 'SOLAR')
95  adf_test(PUN_ALL_15_21_DG['WIND'], 'WIND')
96
97  DIFFERENZIAZIONE
98  PUN_ALL_15_21_DG_diff = PUN_ALL_15_21_DG.diff().dropna()  # .diff()
        esegue la differenziazione, .dropna() rimuove i valori NaN
        risultanti
99
100 for col in ['PUN', 'DEM', 'GAS']:
101     adf_test(PUN_ALL_15_21_DG_diff[col], title=col)
102 print(PUN_ALL_15_21_DG_diff.head())
103 print(PUN_ALL_15_21_DG_diff.tail())
104
105 ADDESTRAMENTO
106 import statsmodels.api as sm
107 model1 = sm.OLS(train, X_train_const)
108 results1 = model.fit()
109 print(results1.summary())
110
111 model2 = sm.OLS(train, X_all_train_const)
112 results2 = model.fit()
113 print(results2.summary())
114
115 CALCOLO PREVISIONI SUL TEST SET
```

```
train_predictions = results.predict(X_train_const)
test_predictions = results.predict(X_test_const)

plt.figure(figsize=(14, 7))
plt.style.use('seaborn-darkgrid')
plt.plot(test.index, test, label='Valori Reali', color='red',
    linewidth=1.5 )
plt.plot(test.index, test_predictions, label='Valori Previsti', color
    ='blue',linewidth=1.5)
plt.legend(fontsize=14)
plt.grid(True)
plt.tight_layout()
plt.xlabel('Data', fontsize=14)
plt.ylabel('PUN [\texteuro/MWh]', fontsize=14)
plt.show()

plt.figure(figsize=(14, 7))
plt.style.use('seaborn-darkgrid')
plt.plot(test.index, test, label='Valori Reali', color='red',
    linewidth=1.5 )
plt.plot(test.index, test_predictions, label='Valori Previsti', color
    ='blue',linewidth=1.5)
plt.legend(fontsize=14)
plt.grid(True)
plt.xlim(pd.Timestamp('2020-04'), pd.Timestamp('2020-05'))
plt.tight_layout()
plt.xlabel('Data', fontsize=14)
plt.ylabel('PUN [\texteuro/MWh]', fontsize=14)
plt.show()

train_all_predictions = results.predict(X_all_train_const)
test_all_predictions = results.predict(X_all_test_const)

plt.figure(figsize=(14, 7))
plt.style.use('seaborn-darkgrid')
plt.plot(test.index, test, label='Valori Reali', color='red',
    linewidth=1.5 )
plt.plot(test.index, test_all_predictions, label='Valori Previsti',
    color='blue',linewidth=1.5)
plt.title('Confronto Train Set: Reali vs Previsti')
plt.legend(fontsize=14)
plt.grid(True)
plt.tight_layout()
plt.xlabel('Data', fontsize=14)
plt.ylabel('PUN [\texteuro/MWh]', fontsize=14)
plt.show()

plt.figure(figsize=(14, 7))
plt.style.use('seaborn-darkgrid')
```

```
159 plt.plot(test.index, test, label='Valori Reali', color='red',
        linewidth=1.5)
160 plt.plot(test.index, test_all_predictions, label='Valori Previsti',
        color='blue',linewidth=1.5)
161 plt.legend(fontsize=14)
162 plt.grid(True)
163 plt.xlim(pd.Timestamp('2020-04'), pd.Timestamp('2020-05'))
164 plt.tight_layout()
165 plt.xlabel('Data', fontsize=14)
166 plt.ylabel('PUN [\texteuro/MWh]', fontsize=14)
167 plt.show()
168
169 VALUTAZIONE DEL MODELLO
170 from math import sqrt
171 test.mean()
172 rmse=np.sqrt(mean_squared_error(test, test_predictions ))
173 print(f'RMSE Score on Test set: {score:0.2f}')
174
175 mae_test = mean_absolute_error(test, test_predictions)
176 print(f"MAE: {mae}")
```

A.4 Extreme Gradient Boosting

```
LIBRERIE
import xgboost as xgb
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from statsmodels.tsa.ar_model import AutoReg

CREAZIONE DUMMIES TEMPORALI
def create_features(ds_15_21):
    """
    Create time series features based on time series index.
    """
    ds_15_21 = ds_15_21.copy()
    ds_15_21['hour'] = ds_15_21.index.hour
    ds_15_21['dayofweek'] = ds_15_21.index.dayofweek
    ds_15_21['quarter'] = ds_15_21.index.quarter
    ds_15_21['month'] = ds_15_21.index.month
    ds_15_21['year'] = ds_15_21.index.year
    ds_15_21['dayofyear'] = ds_15_21.index.dayofyear
    ds_15_21['dayofmonth'] = ds_15_21.index.day
    ds_15_21['weekofyear'] = ds_15_21.index.isocalendar().week
    return ds_15_21

ds_15_21 = create_features(ds_15_21)
ds_15_21

CROSS VALIDATION
from sklearn.model_selection import TimeSeriesSplit

tss = TimeSeriesSplit(n_splits=10, test_size=12*365*1, gap=24)
ds_15_21 = ds_15_21.sort_index()

fig, axs = plt.subplots(10, 1, figsize=(15, 15), sharex=True)
plt.style.use('seaborn-darkgrid')

fold = 0
for train_idx, val_idx in tss.split(ds_15_21):
    train = ds_15_21.iloc[train_idx]
    test = ds_15_21.iloc[val_idx]
    train['PUN'].plot(ax=axs[fold],
                      label='Training Set',
                      title=f'Data Train/Test Split Fold {fold}')
    test['PUN'].plot(ax=axs[fold],
                     label='Test Set')
    axs[fold].axvline(test.index.min(), color='black', ls='--')
```

```
45
46      fold += 1
47  plt.legend()
48  plt.grid(True)
49  plt.tight_layout()
50  plt.show()
51
52  LAG TEMPORALI
53  def add_lags_with_moving_average(ds_15_21):
54      # Mappa originale dei valori di 'PUN'
55      target_map = ds_15_21['PUN'].to_dict()
56
57      # Calcola i lag originali
58      ds_15_21['lag1'] = (ds_15_21.index - pd.Timedelta('364 days')).
        map(target_map)
59      ds_15_21['lag2'] = (ds_15_21.index - pd.Timedelta('728 days')).
        map(target_map)
60      ds_15_21['lag3'] = (ds_15_21.index - pd.Timedelta('1092 days')).
        map(target_map)
61      ds_15_21['lag4'] = (ds_15_21.index - pd.Timedelta('1456 days')).
        map(target_map)
62      ds_15_21['lag5'] = (ds_15_21.index - pd.Timedelta('1820 days')).
        map(target_map)
63      ds_15_21['lag6'] = (ds_15_21.index - pd.Timedelta('2184 days')).
        map(target_map)
64
65      # Calcola i nuovi lag come medie mobili dei 3 lag precedenti
66      ds_15_21['lag7'] = ds_15_21[['lag4', 'lag5', 'lag6']].mean(axis
        =1)
67      ds_15_21['lag8'] = ds_15_21[['lag5', 'lag6', 'lag7']].mean(axis
        =1)
68      ds_15_21['lag9'] = ds_15_21[['lag6', 'lag7', 'lag8']].mean(axis
        =1)
69      ds_15_21['lag10'] = ds_15_21[['lag7', 'lag8', 'lag9']].mean(axis
        =1)
70
71      return ds_15_21
72  ds_15_21 = add_lags_with_moving_average(ds_15_21)
73  ds_15_21
74
75  ADDESTRAMENTO
76  tss = TimeSeriesSplit(n_splits=10, test_size=12*365*1, gap=24)
77  ds_15_21 = ds_15_21.sort_index()
78
79
80  fold = 0
81  preds = []
82  scores = []
83  for train_idx, val_idx in tss.split(ds_15_21):
```

```
84      train = ds_15_21.iloc[train_idx]
85      test = ds_15_21.iloc[val_idx]
86
87      train = create_features(train)
88      test = create_features(test)
89
90      FEATURES = ['DEM', 'SOLAR', 'GAS', 'WIND','dayofyear', 'hour', '
        dayofweek', 'quarter', 'month', 'year', 'lag1','lag2','lag3','lag4
        ', 'lag5','lag6', 'lag7', 'lag8', 'lag9','lag10']
91      TARGET = 'PUN'
92
93      X_train = train[FEATURES]
94      y_train = train[TARGET]
95
96      X_test = test[FEATURES]
97      y_test = test[TARGET]
98
99      reg = xgb.XGBRegressor(base_score=0.5, booster='gbtree',
100                             n_estimators=1000,
101                             early_stopping_rounds=50,
102                             objective='reg:linear',
103                             max_depth=3,
104                             learning_rate=0.01)
105     reg.fit(X_train, y_train,
106             eval_set=[(X_train, y_train), (X_test, y_test)],
107             verbose=100)
108
109     y_pred = reg.predict(X_test)
110     train_predictions = reg.predict(X_train)
111     preds.append(y_pred)
112     score = np.sqrt(mean_squared_error(y_test, y_pred))
113     scores.append(score)
114
115 print(f'Score across folds {np.mean(scores):0.4f}')
116 print(f'Fold scores:{scores}')
117
118 ds_15_21 = create_features(ds_15_21)
119
120 FEATURES = ['DEM', 'SOLAR', 'GAS', 'WIND','dayofyear', 'hour', '
    dayofweek', 'quarter', 'month', 'year', 'lag1','lag2','lag3','lag4
    ', 'lag5','lag6', 'lag7', 'lag8', 'lag9','lag10']
121 TARGET = 'PUN'
122
123 X_all = ds_15_21[FEATURES]
124 y_all = ds_15_21[TARGET]
125
126 reg = xgb.XGBRegressor(base_score=0.5,
127                        booster='gbtree',
128                        n_estimators=500,
```

```
                        objective='reg:linear',
                        max_depth=3,
                        learning_rate=0.01)
reg.fit(X_all, y_all,
        eval_set=[(X_all, y_all)],
        verbose=100)

VALIDAZIONE DEL MODELLO
test_predictions = reg.predict(X_all)

plt.figure(figsize=(14, 7))
plt.plot( ds_15_21.index, y_all, label='Valori Reali', color='blue')
plt.plot( ds_15_21.index, test_predictions, label='Valori Previsti',
    color='red')
plt.title('Prevision on all dataset')
plt.xlabel('Data')
plt.ylabel('PUN')
plt.legend()

start_date = '2020-06-01 00:00:00'
end_date = '2020-06-30 23:00:00'
filtered_index = ds_15_21[start_date:end_date].index

filtered_real_values = y_all[(ds_15_21.index >= start_date) & (
    ds_15_21.index <= end_date)]
filtered_predictions = test_predictions[(ds_15_21.index >= start_date
    ) & (ds_15_21.index <= end_date)]

plt.figure(figsize=(14, 7))
plt.style.use('seaborn-darkgrid')
plt.plot(filtered_index, filtered_real_values, label='Valori Reali',
    color='blue', linewidth=2)
plt.plot(filtered_index, filtered_predictions, label='Valori Previsti
    ', color='red')
plt.title('Previsione sul dataset selezionato', fontsize=16)
plt.xlabel('Data', fontsize=14)
plt.ylabel('PUN', fontsize=14)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

VALUTAZIONE DELLA PERFORMANCE
scoreRMSE = np.sqrt(mean_squared_error(y_all, test_predictions ))
print(f'RMSE Score on Test set: {scoreRMSE:0.2f}')
mse_test = mean_squared_error(y_all, test_predictions )
mae_test = mean_absolute_error(y_all, test_predictions )
r2_test = r2_score(y_all, test_predictions )

```

```
173 CALCOLO PREVISIONI
174 ds_prev = pd.date_range('2021-04-01 00:00:00','2030-03-31 23:00:00',
        freq='1h')
175 ds_21_30 = pd.DataFrame(index=ds_prev)
176 ds_21_30['isFuture'] = True
177 ds_15_21['isFuture'] = False
178 ds_15_30 = pd.concat([ds_15_21, ds_21_30])
179 ds_15_30 = create_features(ds_15_30)
180 ds_15_30_pred = ds_15_30.query('isFuture').copy()
181
182 SCENARIO 1
183 valori_selezionati_dem = ds_15_21.loc['2020-04-01
        00:00:00':'2021-03-31 23:00:00', 'DEM']
184 lunghezza_periodo_dem = len(valori_selezionati_dem)
185 lunghezza_prevista_dem = len(ds_15_30_pred)
186 ripetizioni_necessarie_dem = -(-lunghezza_prevista_dem //
        lunghezza_periodo_dem)
187 num_intervalli = len(ds_15_30_pred)
188
189 pendenza = 0.3
190 incremento_base = 0.0914 # Questo può variare a seconda di come vuoi
        che sia la tua "unità" di incremento
191 retta = [incremento_base * i for i in range(num_intervalli)]
192 ds_15_30_pred['DEM'] += retta
193
194 plt.figure(figsize=(14, 7))
195 plt.plot(ds_15_30_pred.index, ds_15_30_pred['DEM'], color='blue',
        linewidth=2)
196 plt.title('Aumento della domanda di energia nello Scenario 1',
        fontsize=16)
197 plt.xlabel('Data', fontsize=14)
198 plt.ylabel('DEM [MWh]', fontsize=14)
199 plt.legend()
200 plt.grid(True)
201 plt.tight_layout()
202 plt.show()
203
204 valori_solar_ultimo_anno = ds_15_21.loc['2020', 'SOLAR']
205 valori_wind_ultimo_anno = ds_15_21.loc['2018', 'WIND']
206 num_ripetizioni_SOLAR = np.ceil(len(ds_15_30_pred) / len(
        valori_solar_ultimo_anno)).astype(int)
207 num_ripetizioni_WIND = np.ceil(len(ds_15_30_pred) / len(
        valori_wind_ultimo_anno)).astype(int)
208 ds_15_30_pred['SOLAR'] = np.tile(valori_solar_ultimo_anno.values,
        num_ripetizioni_SOLAR)[:len(ds_15_30_pred)]
209 ds_15_30_pred['WIND'] = np.tile(valori_wind_ultimo_anno.values,
        num_ripetizioni_WIND)[:len(ds_15_30_pred)]
210 ds_15_30_pred
211
```

```
212 plt.figure(figsize=(14, 7))
213 plt.style.use('seaborn-darkgrid')
214 plt.plot(ds_15_30_pred.index, ds_15_30_pred['SOLAR'], color='blue',
        linewidth=2)
215 plt.title('Scenario 1', fontsize=16)
216 plt.xlabel('Data', fontsize=14)
217 plt.ylabel('SOLAR [MWh]', fontsize=14)
218 plt.legend()
219 plt.grid(True)
220 plt.tight_layout()
221 plt.show()
222
223 plt.figure(figsize=(14, 7))
224 plt.style.use('seaborn-darkgrid')
225 plt.plot(ds_15_30_pred.index, ds_15_30_pred['WIND'], color='blue',
        linewidth=2)
226 plt.title('Scenario 1', fontsize=16)
227 plt.xlabel('Data', fontsize=14)
228 plt.ylabel('WIND [MWh]', fontsize=14)
229 plt.legend()
230 plt.grid(True)
231 plt.tight_layout()
232 plt.show()
233
234 valori_dem_ultimo_anno = ds_15_21.loc['2020-04-01
        00:00:00':'2021-03-31 23:00:00', 'GAS']
235 num_ripetizioni_DEM = np.ceil(len(ds_15_30_pred) / len(
        valori_dem_ultimo_anno)).astype(int)
236 ds_15_30_pred['GAS'] = np.tile(valori_dem_ultimo_anno.values,
        num_ripetizioni_DEM)[:len(ds_15_30_pred)]
237
238 plt.figure(figsize=(14, 7))
239 plt.style.use('seaborn-darkgrid')
240 plt.plot(ds_15_30_pred.index, ds_15_30_pred['GAS'], color='blue',
        linewidth=2)
241 plt.title('Scenario 1', fontsize=16)
242 plt.xlabel('Data', fontsize=14)
243 plt.ylabel('GAS ', fontsize=14)
244 plt.legend()
245 plt.grid(True)
246 plt.tight_layout()
247 plt.show()
248
249 ds_15_30_pred['PUN_S1'] = reg.predict(ds_15_30_pred[FEATURES])
250
251 plt.figure(figsize=(14, 7))
252 plt.style.use('seaborn-darkgrid')
253 plt.plot(ds_21_30.index, ds_15_30_pred['PUN_S1'], label='Scenario 1',
        color='Orange', linewidth=1)
```

```
254 plt.title('Previsione del PUN - SCENARIO 1', fontsize=16)
255 plt.xlabel('Data', fontsize=14)
256 plt.ylabel('PUN ', fontsize=14)
257 plt.legend()
258 plt.grid(True)
259 plt.tight_layout()
260 plt.show()
261
262 media_mensile_1 = ds_15_30_pred['PUN_S1'].resample('M').mean()
263 mediaANNUALE_1 = ds_15_30_pred['PUN_S1'].resample('Y').mean()
264
265 plt.figure(figsize=(14, 7))
266 media_mensile_1.plot(title='Trend - Scenario 1', label='Media Mensile
    ', color='Blue', linewidth=2)
267 mediaANNUALE_1.plot(title='Trend - Scenario 1', label='Media Annuale
    ', color='Red', linewidth=2)
268 plt.xlabel('Data', fontsize=14)
269 plt.ylabel('PUN ', fontsize=14)
270 plt.legend()
271 plt.grid(True)
272 plt.tight_layout()
273 plt.show()
274
275 SCENARIO 2
276 pendenza = 0.8
277 incremento_base = 0.0048
278 retta = [incremento_base * i for i in range(num_intervalli)]
279
280 # Somma la retta ai valori di SOLAR in ds_15_30_pred
281 ds_15_30_pred['GAS'] += retta
282
283 plt.figure(figsize=(14, 7))
284 plt.plot(ds_15_30_pred.index, ds_15_30_pred['GAS'], color='blue',
    linewidth=2)
285 plt.title('Aumento del prezzo MGP del GAS nello Scenario 2', fontsize
    =16)
286 plt.xlabel('Data', fontsize=14)
287 plt.ylabel('GAS ', fontsize=14)
288 plt.legend()
289 plt.grid(True)
290 plt.tight_layout()
291 plt.show()
292
293 ds_15_30_pred['PUNS2'] = reg.predict(ds_15_30_pred[FEATURES])
294
295 plt.figure(figsize=(14, 7))
296 plt.style.use('seaborn-darkgrid')
297 plt.plot(ds_21_30.index, ds_15_30_pred['PUNS3'], label='Scenario 2',
    color='Orange', linewidth=1)
```

```
298 plt.title('Previsione del PUN - SCENARIO 2', fontsize=16)
299 plt.xlabel('Data', fontsize=14)
300 plt.ylabel('PUN ', fontsize=14)
301 plt.legend()
302 plt.grid(True)
303 plt.tight_layout()
304 plt.show()
305
306 media_mensile_2 = ds_15_30_pred['PUNS3'].resample('M').mean()
307 mediaANNUALE_2 = ds_15_30_pred['PUNS3'].resample('Y').mean()
308
309 plt.figure(figsize=(14, 7))
310 media_mensile_2.plot(title='Trend - Scenario 2', label='Media Mensile
    ', color='Blue', linewidth=2)
311 mediaANNUALE_2.plot(title='Trend - Scenario 2', label='Media Annuale
    ', color='Red', linewidth=2)
312 plt.xlabel('Data', fontsize=14)
313 plt.ylabel('PUN ', fontsize=14)
314 plt.legend()
315 plt.grid(True)
316 plt.tight_layout()
317 plt.show()
318
319 plt.figure(figsize=(14, 7))
320 mediaANNUALE_1.plot(title='Trend - Scenario 1', label='Scenario 1',
    color='Blue', linewidth=2)
321 mediaANNUALE_2.plot(title='Trend - Scenario 2', label='Scenario 2',
    color='Red', linewidth=2)
322 plt.xlabel('Data', fontsize=14)
323 plt.ylabel('PUN ', fontsize=14)
324 plt.legend()
325 plt.grid(True)
326 plt.tight_layout()
327 plt.show()
```

Bibliografia

[1] I. Conti N. Rossetto. «La politica energetica dell'Unione europea: obiettivi e sviluppi dalle origini fino al pacchetto "Energia pulita per tutti gli europei». In: Formazione in materia europea - 2017 (2017), pp. 35–50 (cit. a p. 2).

[2] Consiglio europeo. *Quadro 2030 per il clima e l'energia*. Ottobre 2018. URL: https://ec.europa.eu/clima/policies/strategies/2030_it (cit. a p. 3).

[3] R. Giannetti. «Il servizio elettrico dai sistemi regionali alla liberalizzazione». In: Treccani (mar. 2013), pp. 791–817 (cit. a p. 7).

[4] GME. *Vademecum della Borsa Elettrica Italiana*. Ottobre 2009. URL: https://www.google.com/search?client=firefox-b-d&q=gme+vademecum+borsa+elettrica (cit. a p. 9).

[5] GME. *Vademecum della Borsa Elettrica Italiana*. 2024. URL: https://www.mercatoelettrico.org/it/tools/glossario.aspx (cit. a p. 10).

[6] M. R. Roberts C. R. Knittel. «An empirical examination of restructured electricity prices». In: Energy Economics 27 (nov. 2005), pp. 791–817 (cit. a p. 16).

[7] P. Villaplana A. Escribano J. I. Pena. «Modeling electricity price: international evidence». In: Working Paper 02-27 (nov. 2002), pp. 1–32 (cit. a p. 16).

[8] G. Hu L. Jiang. «A Review on Short-Term Electricity Price Forecasting Techniques for Energy Markets». In: Energy Economics 27 (nov. 2018) (cit. a p. 18).

[9] GME. *Statistiche*. URL: www.mercatoelettrico.org/it/Statistiche (cit. alle pp. 24, 32).

[10] ENTSO-E. *Transparency Plactform*. URL: https://transparency.entsoe.eu (cit. a p. 27).

[11] e George Athanasopoulos Hyndman Rob J. «Forecasting: Principles and Practice.» In: 2nd ed OTexts (nov. 2018) (cit. a p. 36).

[12] David S. Stoffer Shumway Robert H. «Time Series Analysis and Its Applications: With R Examples.» In: 4th ed. Springer (2017) (cit. a p. 36).

[13] Carlos Guestrin Tianqi Chen. «XGBoost: A Scalable Tree Boosting System.» In: (2016) (cit. a p. 67).

[14] Cha Zhang. «Ensemble Machine Learning: Methods and Applications». In: *2012th Edition*, p. 5 (cit. a p. 68).

[15] SNAM TERNA. «Documento di Descrizione degli Scenari 2022». In: *EXECUTIVE SUMMARY*. Gen. 2022 (cit. a p. 79).