POLITECNICO DI TORINO

Collegio di Ingegneria Gestionale e della Produzione

Corso di Laurea Magistrale

in Ingegneria Gestionale

Tesi di Laurea Magistrale

Previsione long-term di prezzo di mercato elettrico

Relatori

prof. Maurizio Repetto

prof. Paolo Lazzeroni

Candidato

Pietro Simeone

Dicembre 2023

Indice

Introduzione 1
1. Transizione energetica 3
1.1 Contesto 4
1.2 Il mercato elettrico italiano 4
1.3 Il bilanciamento della rete 5
2. Creazione del dataset 7
2.1 Analisi del periodo scelto 7
2.2 Struttura del dataset 7
2.3 Introduzione a Python 8
2.4 Librerie di Python utilizzate 10
2.5 Creazione del dataset su Python 11
3. Variabili dummy ed analisi di regressione 12
3.1 Le variabili dummy 13
3.2 Analisi di regressione 16
3.2 La regressione lineare applicata al caso studio 19
3.3 Il metodo dei minimi quadrati 21
3.4 Applicazione del metodo dei minimi quadrati al caso studio 25
4. Risultati del modello 27
4.1 Analisi dei risultati e grafici di confronto 27
4.2 Ultimo appunto sull'errore prodotto dal modello 36
5. Conclusioni 38
Bibliografia 39
Appendice 42

Introduzione

Il consumo di energia elettrica aumenta costantemente, non solo a causa dell'aumento della popolazione globale e da abitudini sempre più elettrico-centriche ma anche a causa dell'avvento dell'auto elettrica che sta prepotentemente facendo il suo ingresso sul mercato.

Secondo l'articolo scritto da Markus Wacket e pubblicato sulla rivista "Reuters", l'Europa ha imposto che dal primo gennaio 2035 non potranno più essere commercializzate auto che hanno come propulsione dei combustibili fossili ma solo auto elettriche e per un po' di anni anche ibride.

Questa scelta deriva sia dal desiderio di imporsi come leader mondiali nel settore delle auto elettriche e quindi di essere all'avanguardia nel settore, ma anche e soprattutto per abbattere le emissioni di combustibili fossili che sono alla base del cambiamento climatico di cui sempre di più ci stiamo rendendo conto.

Quindi entro il 2035 tutti possederanno un'auto elettrica, diventerà il mezzo comune di spostamento e perfino i mezzi pubblici avranno motori elettrici.

A causa del cambiamento climatico le temperature aumenteranno nel prossimo decennio di circa 1,5 gradi (Fonte: IPCC – Intergovernmental Panel on Climate Change), nel caso più ottimistico, rendendo le stagioni sempre più calde e costringendo le popolazioni ad un utilizzo ancora più intensivo degli apparecchi di ventilazione e condizionamento.

Può sembrare una cosa di poco conto ma questi apparecchi hanno un forte effetto sulla domanda di energia elettrica e conseguentemente ad un aumento della popolazione mondiale ed un aumento delle temperature globali ce ne sarà sempre più bisogno.

Questi sono solo alcuni esempi di fattori che porteranno ad un sensibile aumento della domanda di energia elettrica.

Qui si pone un problema, cercando di diventare sempre più ecosostenibili (basti pensare al caso delle auto elettriche) non si può continuare a produrre energia da combustibili fossili, si punta sempre al diventare più "green".

La soluzione sono le energie rinnovabili che permettono di produrre energia elettrica sfruttando le fonti che il pianeta Terra mette a disposizione naturalmente.

Le energie rinnovabili presentano molti vantaggi, tra cui il poter disporre di energia "pulita" e l'avere un costo marginale di produzione relativamente basso.

Tutti si soffermano sugli aspetti positivi e di quanto le energie rinnovabili siano importanti e fondamentali per il nostro futuro e di conseguenza per la nostra sopravvivenza; però non molti si soffermano sugli aspetti negativi di questa tecnologia.

Gli aspetti negativi sono due in particolare:

1. La penetrazione dell'energia rinnovabile nel sistema elettrico crea uno squilibrio di potenza nel sistema dato che le fonti rinnovabili non sono spesso programmabili.

2. Altro aspetto che si lega un po' al precedente, non avendo a disposizione tecnologie e mezzi per lo stoccaggio di energia elettrica ci sarebbe un problema di disponibilità di energia elettrica in quanto, nel caso dei pannelli solari, ad esempio, la produzione di energia sarebbe possibile solo in presenza di luce solare quindi sarebbe alquanto imprevedibile.

Per quanto riguarda questi due punti si comprende benissimo che non si potrà fare affidamento solo sulle fonti rinnovabili ma si necessiterà di impianti di produzione che possano da un lato essere utili per il dispacciamento dell'energia elettrica e permettere di rendere la rete più stabile.

Altro aspetto, da tenere in considerazione ovvero ad un aumento di penetrazione delle energie rinnovabili nel sistema di produzione elettrica corrisponde una proporzionale diminuzione del prezzo di equilibrio dell'energia elettrica, ciò a causa dell'equilibrio domanda-offerta.

Aumentando la produzione da rinnovabili, diminuisce il prezzo dell'energia elettrica e di conseguenza aumenta anche il tempo per il ritorno di un investimento in un impianto di produzione da fonti rinnovabili.

Quindi si arriverà ad un punto in cui non converrà più creare nuovi impianti in quanto il tempo di ritorno dell'investimento sarebbe troppo elevato.

Un ruolo chiave nel processo di creazione di nuovi impianti ce l'ha lo Stato e l'Europa fornendo incentivi per la loro realizzazione.

Nonostante questo effetto delle rinnovabili, il prezzo dell'energia elettrica segue comunque una certa stagionalità, infatti in base al periodo dell'anno in cui ci troviamo avrà un prezzo più o meno alto.

È proprio questo l'obiettivo di questa tesi ovvero prendendo i dati relativi al giorno dell'anno e al relativo PUN da un dataset presente in un file Excel (capitolo 1), costruire una matrice di variabili dummy (capitolo 2) sulla quale verrà effettuata una analisi dei minimi quadrati (capitolo 3) per trovare i coefficienti del PUN legati alla stagionalità che ci permettano in qualche modo di captare l'andamento dei prezzi dell'energia elettrica durante le varie stagioni.

Il PUN (Prezzo Unico Nazionale) sarebbe il prezzo a cui acquista l'utente finale ed è "la media pesata nazionale dei prezzi zonali di vendita dell'energia elettrica per ogni ora e per ogni giorno" (Fonte: Gestore dei Servizi Energetici GSE).

Le serie temporali del PUN sono facilmente reperibili sul sito del GME (Gestore dei Mercati Elettrici) ed è proprio da questa fonte che sono stati scaricati tutti i valori orari dal 2005 ad oggi utili per l'analisi e la creazione del dataset.

Mentre le variabili dummy sono, sostanzialmente, delle variabili che, in base ad una determinata condizione o ad un insieme di condizioni, possono assumere due soli valori ovvero "1" per indicare il valore "True", cioè la condizione è rispettata, oppure "0" per indicare il valore "False" per non rispettata.

Tutto questo studio è stato fatto facendo uso del linguaggio di programmazione Python ed in particolare utilizzando le librerie di Pandas, NumPy e Matplotlib.

1. Transizione energetica

Il mercato dell'energia è un ambiente complesso ed in esso il prezzo dell'energia elettrica viene determinato attraverso un sistema ad asta.

La determinazione di questo prezzo viene ricavato dalla intersezione della curva di domanda (S) e della curva di offerta (D) di energia elettrica (Figura 1).

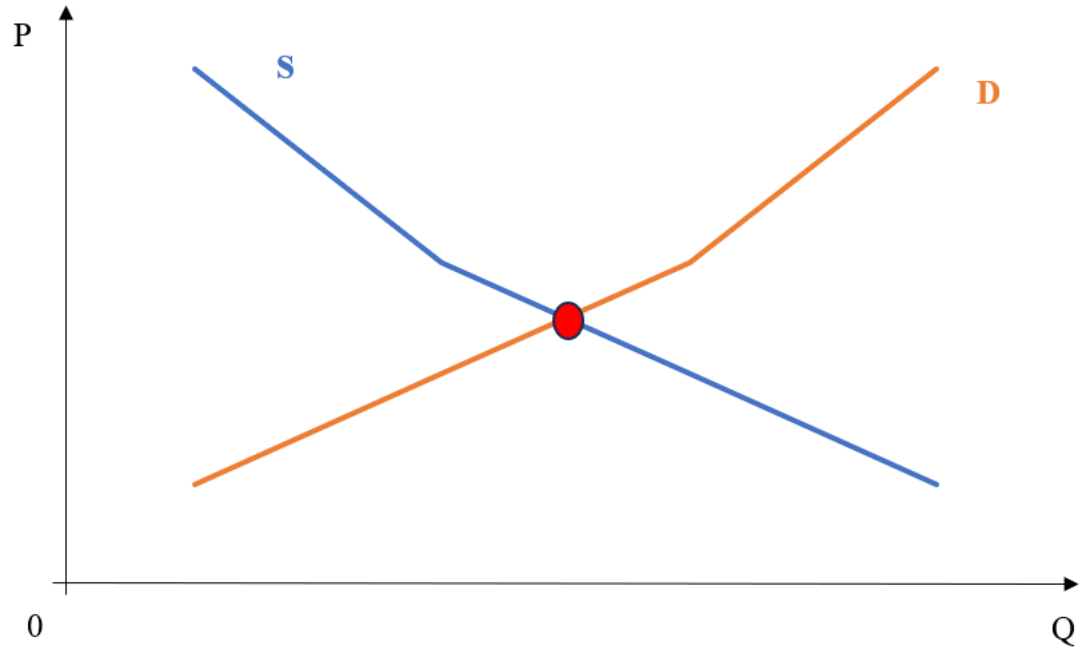

Figura 1: Curva di domanda ed offerta di energia elettrica

In rosso c'è il punto di equilibrio che determina il prezzo dell'energia elettrica e si può ben vedere che, lasciando invariata l'offerta di energia elettrica, ad un aumento di domanda corrisponde un aumento di prezzo; viceversa, lasciando invariata la domanda di energia elettrica, ad un aumento di offerta corrisponde una diminuzione di prezzo.

In questo capitolo verrà affrontato il tema di come viene determinato il prezzo dell'energia elettrica e di alcune implicazioni derivanti dal passaggio da un sistema basato su impianti di produzione elettrica con combustibili fossili, ad uno con presenza di produzione elettrica da fonti rinnovabili.

1.1 Contesto

Il mercato dell'energia elettrica ha visto una crescente penetrazione delle fonti di energia rinnovabili nel sistema produttivo.

Basti pensare al fotovoltaico che è passato da essere fornitore di energia elettrica per una percentuale prossima allo 0 % nel 2010 ad una percentuale, nelle ore diurne, del 2023 pari a circa il 20% (Fonte: Terna).

Questo aumento sproporzionato è diretta conseguenza di due politiche:

- POLITICHE NAZIONALI
- POLITICHE EUROPEE

Per quanto concerne le politiche nazionali si riferiscono al meccanismo di incentivi, creato dallo Stato italiano, per favorire la creazione di impianti di generazione elettrica da fonti rinnovabili e che ha portato anche utenti privati ad installare pannelli fotovoltaici e/o pale eoliche.

Le politiche europee invece si riferiscono alla direttiva del Consiglio Europeo il "Green Deal" che si pone come obiettivo quello della riduzione delle emissioni di CO2, entro il 2030, del 55% e per raggiungere il suddetto obiettivo è stato inoltre imposto che ogni Paese abbia il 45 % dell'energia prodotta derivante da fonti rinnovabili (Fonte: "European Commission").

Tutto ciò è stato deciso per ridurre l'inquinamento e quindi contrastare o perlomeno limitare l'effetto del cambiamento climatico e l'aumento delle temperature globali.

Questo aumento della produzione da fonti rinnovabili ha effetto non solo sul prezzo dell'energia elettrica ma sul sistema elettrico in generale.

Prima di entrare nel merito di questi effetti, è doveroso soffermarsi su come funziona il mercato elettrico dell'Italia, quali sono le sue sessioni e come si compongono.

1.2 Il mercato elettrico italiano

Il prezzo finale dell'energia elettrica che il consumatore vede in bolletta è frutto del risultato di tre sessioni di contrattazione che sono:

1. MERCATO DEL GIORNO PRIMA (MGP)
2. MERCATO INTRA-DAY (MI) o MERCATO INTERGIORNALIERO
3. MERCATO DEI SERVIZI AUSILIARI

Lo studio si concentrerà sul mercato del giorno prima (MGP) e il mercato dei servizi ausiliari che sono di interesse maggiore.

Il mercato del giorno prima apre 9 giorni prima rispetto al giorno di consegna e chiude il giorno prima rispetto alla consegna.

Funziona proprio come una borsa in cui il prodotto di scambio sono i blocchi orari del giorno successivo; queste operazioni di acquisto/vendita avvengono sotto la supervisione del GME (Gestore dei Mercati Elettrici).

Questo scambio viene fatto in tutte le zone geografiche in cui è diviso il territorio italiano e la media ponderata dei prezzi di scambio delle varie zone è il PUN (Prezzo Unico Nazionale).

Le fonti di energia rinnovabile hanno la priorità nella fornitura di energia elettrica in quanto inquinano meno e ciò porta, come detto in precedenza, ad un aumento di offerta di energia elettrica e ciò porta ad una diminuzione del prezzo di equilibrio e quindi una diminuzione del prezzo dell'energia elettrica.

Un altro fattore da considerare è il costo marginale delle diverse fonti energetiche ed il concetto di prezzo marginale.

Le fonti rinnovabili non dovendo acquistare materie prime per la produzione di energia elettrica, presentano come unici costi quelli operativi, ragion per cui presentano un costo marginale di produzione decisamente inferiore rispetto alle fonti tradizionali.

Si potrebbe pensare che ciò porti ad una diminuzione del prezzo dell'energia ma ciò non è completamente vero a causa del sistema di determinazione del prezzo dell'energia elettrica detto "sistema di prezzo marginale dell'energia".

Il sistema funziona in questo modo:

- I produttori di energia elettrica dichiarano quanta energia potrebbero produrre il giorno successivo per tutte le ore del giorno ed a quale prezzo (il cosiddetto MGP).
- Il Gestore dei Mercati Energetici (GME) ed inizia a raccogliere queste offerte partendo da coloro che hanno presentato le offerte con prezzo minore (le fonti rinnovabili) fino a passare a quelle con un prezzo maggiore (solitamente offerte pervenute da produttori di energia elettrica da centrali a gas), fino a che non si raggiunge la stima di fabbisogno energetico per il giorno successivo.
- Infine, il prezzo di equilibrio dell'energia elettrica detto anche "prezzo marginale" è pari a quello relativo all'ultima offerta accettata quindi il prezzo maggiore proposto.

Ciò porta le aziende che si occupano di generazione di energia elettrica da fonti tradizionali a fare meno profitti sul mercato del giorno prima (MGP).

Questo porta ad un quesito, come fanno queste aziende a recuperare i profitti persi?

Seppur non riescono a compensare totalmente le perdite ottenute nel mercato del giorno prima, riescono a recuperare parte dei profitti con i servizi di bilanciamento della rete.

1.3 Il bilanciamento della rete

È stato detto che nell'MGP vengono scambiati blocchi orari per il giorno successivo, ciò significa che vengono scambiate quantità di energia elettrica; inoltre è stato detto che le fonti rinnovabili hanno la priorità sulla fornitura di energia elettrica.

Quindi che le unità produttive di energia elettrica da fonti di rinnovabili assicurano di fornire un determinato quantitativo di energia in un giorno futuro, il che è impensabile.

Infatti, è impossibile calcolare con certezza la quantità di energia prodotta con il rinnovabile poiché questa deriva da fonti legate al pianeta Terra che sono di per sé imprevedibili (basti pensare alle previsioni meteorologiche).

Oltre ciò le rinnovabili hanno natura intermittente, infatti, prendendo come esempio il fotovoltaico, si ha che la produzione è concentrata nelle ore diurne ed assente nelle ore serali e perciò questa fonte, nelle ore senza luce solare, esce dal mercato della produzione.

Si capisce bene che ci sono due problemi principali:

- La quantità di energia che si produrrà in una data futura è imprevedibile
- Le rinnovabili hanno natura intermittente e non si può fare completo affidamento su esse

Ipotizzando che la domanda di energia elettrica rimanga pressoché costante, si viene a creare uno squilibrio tra domanda e offerta determinando uno squilibrio di frequenza sulla rete, che può portare a seri problemi sull'intero sistema elettrico.

Per far fronte a ciò, all'interno del mercato dei servizi ausiliari, c'è il mercato dei servizi di bilanciamento (MB) in cui partecipano le tecnologie di produzione tradizionale e che permettono di colmare questo gap tra domanda e offerta immettendo (o rimuovendo) dal sistema elettrico quantitativi di energia.

Data la richiesta di voler ammettere più fonti di produzione elettrica al mercato del bilanciamento, da fine 2018 Terna ha avviato dei progetti pilota per ammettere a questo mercato del UVAM (Unità Virtuali Abilitate Miste) e sono costituite da unità di consumo, di produzione e di accumulo (Fonte: "Le cinque cose da sapere quando si parla di UVAM" lightbox).

Queste possono modificare la loro produzione e consumo rappresentando un impianto di generazione/consumo virtuale.

Le UVAM, quindi, permettono anche alle fonti di generazione di energia da fonti rinnovabili di partecipare al mercato dei servizi di bilanciamento.

Questo servizio viene effettuato da Terna la quale, essendo responsabile della sicurezza del sistema elettrico, monitora in tempo reale la frequenza del sistema e svolge due operazioni dette "chiamata a salire" oppure "chiamata a scendere" con cui regola la frequenza del sistema facendo in modo che la quantità di potenza immessa in rete sia uguale a quella consumata.

In precedenza, è stato detto che è in questa sessione che le aziende con impianti di produzione con fonti tradizionali recuperano parte dei profitti, infatti esse vengono remunerate per questo servizio di bilanciamento ed essendo relativamente poche le aziende ammesse a questo "mercato" hanno alto potere di mercato e possono richiedere un prezzo decisamente alto per fare queste operazioni.

Questo potere di mercato nelle sessioni di bilanciamento porta queste aziende a trattenere capacità nelle sessioni dell'MGP e rilasciarla nel mercato del bilanciamento dove ottengono profitti decisamente maggiori.

Tutto questo pesa sulla bolletta dei consumatori perché il risparmio che avrebbero con la produzione da rinnovabili si perde con i costi necessari al bilanciamento della rete

2. Creazione del dataset

La creazione di un dataset preciso e affidabile è uno degli elementi principali della analisi dati, specialmente se si lavora con dati di una certa complessità come i prezzi dell'energia elettrica (PUN).

In questo capitolo si affronterà la creazione del dataset su Excel che servirà in seguito per le analisi con Python.

In particolare, i dati di interesse sono quelli relativi alla data intesa come: Anno, Mese, Giorno del Mese, Giorno della Settimana, all'ora del giorno (dall'1 alle 24) ed infine il PUN per ogni ora del giorno.

2.1 Analisi del periodo scelto

Prima di parlare di come è stato costruito il dataset si analizzerà il periodo preso in considerazione.

Sono stati utilizzati i dati dal 2005 al 2010 per cercare di annullare l'effetto delle rinnovabili sul PUN ed avere un dato del prezzo dell'energia elettrica che sia privo di questo effetto.

L'effetto delle rinnovabili sul PUN sarebbe che ad un aumento della produzione di energia elettrica da fonti rinnovabili consegue una proporzionale diminuzione del PUN, come detto in precedenza per l'equilibrio domanda-offerta.

2.2 Struttura del dataset

La struttura del dataset (Figura 2) è composta da sei colonne distinte:

1. Anno, dal 2005 a 2010
2. Mese (ad es. Gennaio)
3. Giorno del mese, numero che assume un valore da 1 a 31
4. Giorno della settimana, dal lunedì alla domenica
5. Ora, da 1 a 24
6. Valore del PUN in €/Mwh

Anno	Mese	Giorno_Mese	Giorno_Settimana	Ora	PUN
2005	Gennaio	1	Sabato	1,00	27,65
2005	Gennaio	1	Sabato	2,00	24,09
2005	Gennaio	1	Sabato	3,00	24,50
2005	Gennaio	1	Sabato	4,00	23,96
2005	Gennaio	1	Sabato	5,00	24,47
2005	Gennaio	1	Sabato	6,00	23,90
2005	Gennaio	1	Sabato	7,00	23,90
2005	Gennaio	1	Sabato	8,00	23,97
2005	Gennaio	1	Sabato	9,00	16,10
2005	Gennaio	1	Sabato	10,00	10,71
2005	Gennaio	1	Sabato	11,00	11,29

Figura 2: Dataset Excel 2005_2010

In questa fase il set di dati è ancora "grezzo" questo significa che non può essere utilizzato subito ma necessita di una pulizia.

Pulire un dataset significa controllare che il dataset non presenti valori mancanti, celle vuote oppure valori che devono essere eliminati.

Per prima cosa si verifica che non ci siano celle vuote e per questo set di dati non ce ne erano.

Seconda cosa si verifica che non ci siano dati mancanti e questo problema si è presentato poiché non erano presenti dati per alcune ore; quindi, è stata inserita la riga mancante e per il dato del PUN è stata fatta la media tra il dato precedente e quello successivo.

Terza cosa si verifica che non ci siano valori da eliminare e per questa verifica è apparso un valore di ora "25" che è stato prontamente eliminato.

Tutta questa operazione di pulizia è stata fatta utilizzando Excel e delle sue funzioni, come la funzione "se" o anche il filtro tabelle; soprattutto quest'ultimo strumento è stato fondamentale perché ha permesso di analizzare il dataset velocemente e con precisione.

Fatto ciò, il dataset è pronto per essere richiamato su Python per l'analisi.

2.3 Introduzione a Python

Questa analisi inizialmente non era effettuata quasi interamente grazie all'utilizzo di Python ma molte operazioni erano effettuate su Excel.

Le problematiche del seguire questo approccio erano molteplici, in primis ogni volta che si voleva fare una analisi, ad esempio cambiando il periodo di riferimento, bisognava o modificare un file esistente o crearne uno nuovo.

Questo ha portato, ad un certo punto, ad avere molti file Excel e ciò rendeva molto confusionario il ritrovare il file di cui si necessitava.

Altro elemento è la dimensione dei file, infatti essendo le operazioni effettuate su un grande quantitativo di dati e necessitando anche di molte colonne, la dimensione dei file era di centinaia di Megabyte rendendo, la semplice apertura del file, un'operazione che richiedeva un tempo non indifferente ed uno sforzo considerevole per il computer.

Ultimo elemento è l'errore casuale dovuto appunto a fenomeno associabili al caso e perciò non controllabili.

Lavorando direttamente sul file Excel e dovendo eseguire operazioni ripetitive, la probabilità di errore era abbastanza elevata e ciò forniva non poche perplessità quando si dovevano analizzare i risultati; non si sapeva se fossero presenti degli errori casuali che potevano aver condizionato la buona riuscita delle operazioni.

Per tutte queste ragioni è stato scelto di utilizzare un dataset iniziale in cui sono stati semplicemente inseriti i valori estratti dal sito del GME e poi tutto il resto è stato fatto direttamente su Python.

Python è stato scelto sia perché ha reso queste operazioni di modifica del dataset più semplici, non dovendo ogni volta creare nuovi file Excel e sia perché essendo un linguaggio di alto livello presenta una maggiore semplicità di comprensione e di conseguenza di applicazione al caso rispetto ad altri linguaggi di programmazione come ad esempio il C.

Il codice viene eseguito direttamente dopo aver scritto una riga ed aver premuto il tasto "invio", quindi è molto facile capire dove è l'errore poiché Python lo segnala subito e ciò permette di scrivere un codice perfettamente funzionante avendo la possibilità di controllare man mano che il codice inserito sia corretto.

L'assenza di parentesi graffe (utili, ad esempio, per identificare il contenuto di un ciclo for) e la sostituzione di esse con l'indentazione, che sarebbe il lasciare dello spazio rispetto al bordo sinistro per delineare il contenuto di un ciclo for o un ciclo if, rende Python molto leggibile e rimuove la possibilità di errore per mancanza di parentesi graffa.

Ultimo aspetto, ma non ultimo per importanza, è la presenza delle librerie.

Le librerie non sono altro che un contenitore, che può essere richiamato in Python, al cui interno ci sono delle funzioni per effettuare delle operazioni.

Queste funzioni presentano una struttura molto più semplificata rispetto al dover scrivere il codice da zero e perciò sono abbastanza facili da utilizzare e da comprendere.

Attraverso la parola "import" si possono appunto importare tutte le librerie necessarie per l'analisi all'interno del proprio editor di codice; attenzione perché senza questa operazione di import non sarà possibile utilizzare le funzioni presenti in una data libreria.

Prima di poter fare l'import e successivamente usare le funzioni delle librerie, esse devono essere installate sull'editor che si sceglie di utilizzare.

L'editor utilizzato nelle analisi è "Visual Studio 2022" che presenta una certa semplicità di utilizzo ed una interfaccia più user-friendly rispetto all'editor di Python 3.10 scaricabile dal sito ufficiale (www.python.org).

2.4 Librerie di Python utilizzate

Si inizia visualizzando quali sono le librerie utilizzate:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

È stato già detto in precedenza che "import" ti permette di aggiungere delle librerie al tuo codice mentre con "as" si definiscono dei caratteri per richiamare una funzione presente in quella libreria.

Le librerie utilizzate sono 3:

- Pandas
- NumPy
- Matplotlib

Vediamo nel dettaglio cosa sono e cosa ci permettono di fare.

La prima di queste librerie (forse anche la più importante) è Pandas essa viene utilizzata nell'ambito della data science e del Machine Learning permettendo sia di effettuare analisi di dati e sia di creare modelli predittivi.

Pandas lavora principalmente con DataFrame che possono essere delle matrici o tabelle in due dimensioni e su di esse permette di effettuare tutta una serie di operazioni che potrebbero essere similmente effettuate su Excel.

Questa libreria permette anche il caricamento e salvataggio di dati, ad esempio da file Excel.

In definitiva, Pandas è uno strumento fondamentale per l'analisi dati su Python e permette di effettuare diverse operazioni in maniera abbastanza semplice e utilizzando funzioni dalla logica intuitiva.

La seconda libreria è NumPy, anch'essa viene utilizzata nella data science e nel machine learning, la differenza è che con questo pacchetto puoi effettuare le operazioni tra vettori, tra matrici o tra vettore e matrice; operazioni che possono essere: somma, sottrazione, moltiplicazione e divisione.

In generale, NumPy è molto utile per eseguire delle operazioni matematiche avanzate principalmente su vettori (arrays) ma non solo.

Terza ed ultima libreria è Matplotlib, che è di supporto alle altre due poichè permette di visualizzare i risultati delle analisi effettuate sulla matrice su dei grafici così da essere più facilmente visibili.

Quindi Matplotlib permette di creare grafici (non solo grafici ma anche altre tipologie di immagini) di alta qualità basandosi su dei dati (lavora spesso con le librerie Pandas e NumPy) per una comunicazione visiva dei risultati di analisi e permette inoltre di personalizzare a pieno il grafico cambiando colori, titoli, etichette, ecc. per adattarlo alle proprie esigenze e necessità.

2.5 Creazione del dataset su Python

Dopo aver importato le librerie utili per l'analisi si può passare alla creazione del DataFrame.

Si parte assegnando il percorso del file Excel contenente tutti i valori necessari a una variabile di tipo stringa chiamata "file_path".

```
file_path = 'C:\\Users\\ASUS\\Desktop\\PREZZI ENERGIA\\PUN 2005_2010.xlsx'
```

Ora, utilizzando la libreria Pandas, si legge il file Excel e si ottiene un DataFrame, chiamato "df", contenente i valori presenti in quel file e ciò ci sarà utile per effettuare tutte le operazioni.

```
df = pd.read_excel(file_path)
```

Adesso si crea una copia di questo set di dati, chiamandolo "data", in modo tale da poter lavorare su una copia del dataset senza dover modificare il file Excel sorgente.

```
data=df.copy()
```

Fatte tutte queste operazioni, il DataFrame (Figura 3) è pronto e si può passare alla fase di creazione delle variabili dummy.

	Anno	Mese	Giorno_Mese	Giorno_Settimana	Ora	PUN
0	2005	Gennaio	1	Sabato	1	27.650421
1	2005	Gennaio	1	Sabato	2	24.088555
2	2005	Gennaio	1	Sabato	3	24.499444
3	2005	Gennaio	1	Sabato	4	23.960878
4	2005	Gennaio	1	Sabato	5	24.467464
...	...	...	...	...	...	...
52579	2010	Dicembre	31	Venerdì	20	65.046583
52580	2010	Dicembre	31	Venerdì	21	57.579383
52581	2010	Dicembre	31	Venerdì	22	55.159159
52582	2010	Dicembre	31	Venerdì	23	52.400000
52583	2010	Dicembre	31	Venerdì	24	54.640000

Figura 3: DataFrame

3. Variabili dummy ed analisi di regressione

Ora che si ha a disposizione il dataset si può entrare nel vivo della analisi della stagionalità sul prezzo dell'energia elettrica.

Sono stati utilizzati per questo calcolo i dati relativi al mese dell'anno, al giorno della settimana ed all'ora del giorno e con questi è stata costruita la matrice delle variabili dummy.

Per capire il perché sono necessarie le variabili dummy, vengono mostrati due grafici uno relativo alla media su 5 anni del PUN mensile (Figura 4) ed uno relativo alla media su 5 anni del PUN per ogni giorno della settimana (Figura 5).

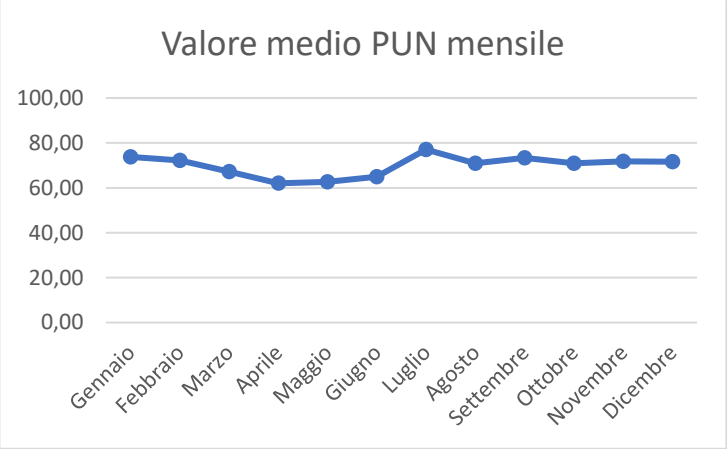

Figura 4: valore medio del PUN per ogni mese dal 2005 al 2010

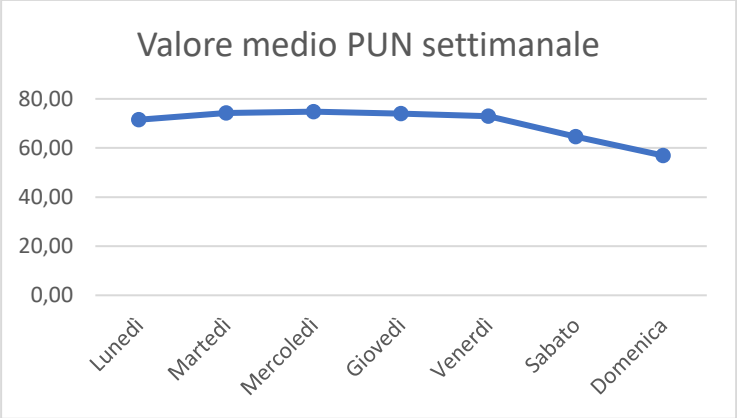

Figura 5: valore medio del PUN per ogni giorno della settimana dal 2005 al 2010

Si può vedere che c'è una certa tendenza nel PUN in base al periodo in cui ci si trova; le variabili dummy permettono di rappresentare un dato come può essere un mese dell'anno e ciò è utile per andare ad identificare e replicare il trend rappresentato.

Nell'analisi dei dati c'è bisogno di utilizzare dei dati numerici per effettuare le analisi; infatti, per le analisi di regressione che verranno effettuate in seguito ci sarebbero dei problemi nella esecuzione se le variabili fossero non numeriche.

Quindi quando si ha a che fare con queste ultime bisogna cercare di modificarle per fare in modo che il concetto contenuto ad ognuna di essa sia racchiusa in un numero.

Ci sono diverse metodologie per fare questa operazione, una di queste è l'utilizzo delle variabili dummy.

3.1 Le variabili dummy

Le variabili dummy vengono utilizzate nel campo dell'analisi dei dati quando si ha a che fare con un dato non numerico, quindi qualitativo (come può essere ad esempio il giorno della settimana).

Bisogna prima di tutto definire la condizione a cui fa riferimento la variabile dummy, ad esempio "è il mese di gennaio?", ed a questa condizione sono associati due stati: lo stato di "True" (rappresentato dall'1) e lo stato di "False" (rappresentato dallo 0).

"True" se la condizione associata a quella variabile è verificata, mentre "False" se non lo è.

In questo studio sono state utilizzate oltre 2000 variabili dummy però esse non solo semplicemente raggruppate in mesi, giorni della settimana e ore del giorno ma sono state strutturate in modo che ognuna di essa contenga all'interno tutte e 3 queste informazioni; quindi, si ha una variabile per ogni combinazione di questi elementi, ad esempio avremo una dummy per l'ora 1 del lunedì di gennaio.

È stato scelto di operare in questa maniera per ottenere una misurazione che sia la più precisa possibile e che permetta di fare analisi più accurate.

Questa operazione di creazione delle dummies è stata fatta mediante l'utilizzo della libreria pandas che permette di gestire e modificare tabelle in maniera semplice e rapida.

Ora si analizza il codice utilizzato in Python.

data['Comb']=data['Ora'].astype(str) + '_' + data['Giorno_Settimana'] + '_' + data['Mese']

Con questa riga di codice si crea una nuova colonna all'interno del DataFrame, chiamata "Comb" (Figura 6), che è combinazione delle tre colonne esistenti ovvero: "Ora", "Giorno_Settimana" e "Mese", inoltre la colonna "Ora" viene convertita in una stringa.

	Comb
0	1_Sabato_Gennaio
1	2_Sabato_Gennaio
2	3_Sabato_Gennaio
3	4_Sabato_Gennaio
4	5_Sabato_Gennaio
...	...
52579	20_Venerdì_Dicembre
52580	21_Venerdì_Dicembre
52581	22_Venerdì_Dicembre
52582	23_Venerdì_Dicembre
52583	24_Venerdì_Dicembre

Figura 6: Colonna "Comb"

Alla fine, ogni riga di questa colonna avrà un valore che combina l'ora con il mese ed il giorno della settimana ed il formato sarà il seguente:

Ora_GiornoSettimana_Mese

Fatto ciò, si inizia con la creazione delle variabili dummy.

dummy_ora=pd.get_dummies(data['Comb'])

Con questa si creano le dummies utilizzando la funzione "get_dummies" di pandas facendo riferimento alla colonna "Comb" creata in precedenza e crea queste variabili basandosi sui diversi valori presenti in questa colonna.

Quindi con "get_dummies" verranno create nuove colonne per ogni combinazione unica di "Ora", "Mese" e "Giorno_Settimana" e darà come risultato "True" se la combinazione è presente e "False" se non lo è.

"dummy ora" è un nuovo DataFrame ma si vuole ottenere un dataset unico, quindi, è necessario aggiungere le colonne create con la funzione "get_dummies" al DataFrame "data":

data=pd.concat([data,dummy_ora], axis=1)

Ogni colonna aggiunta a questo dataset “data” rappresenta una combinazione unica di valori della colonna “Comb”.

Il riferimento “axis=1” serve per comunicare a Python di concatenare i due dataframe, aggiungendo “dummy_ora” a “data”, lungo l’asse delle colonne.

Considerando che per la analisi si necessita dati numerici è necessario convertire i risultati delle dummies ovvero i “True” e “False” in formato numerico in questo caso con 1.0 e 0.0:

data = data.replace({True: 1.0, False: 0.0})

	Anno	Mese	...	9_Venerdì_Ottobre	9_Venerdì_Settembre
0	2005	Gennaio	...	0.0	0.0
1	2005	Gennaio	...	0.0	0.0
2	2005	Gennaio	...	0.0	0.0
3	2005	Gennaio	...	0.0	0.0
4	2005	Gennaio	...	0.0	0.0
...	...	...	...	...	...
52579	2010	Dicembre	...	0.0	0.0
52580	2010	Dicembre	...	0.0	0.0
52581	2010	Dicembre	...	0.0	0.0
52582	2010	Dicembre	...	0.0	0.0
52583	2010	Dicembre	...	0.0	0.0

Figura 7: Dataset con variabili dummy

Ora che è stata completata la creazione della matrice di variabili dummies, si ha a disposizione un dataset pronto (Figura 7) per la fase successiva che è quella della analisi di regressione dei minimi quadrati.

3.2 Analisi di regressione

Per studiare la relazione tra due o più variabili il metodo più utilizzato è quello della regressione lineare che può essere semplice oppure multipla.

Nei modelli di regressione è sempre presente una variabile dipendente (Y) detta anche variabile di risposta quantitativa ed una o più variabili indipendenti (X) dette regressori.

Se si è in presenza di regressione lineare semplice, si avrà una solo regressore e quindi l'analisi verrà fatta con due variabili; mentre con la regressione lineare multipla si avranno più regressori e l'analisi verrà fatta con n variabili.

Il termine "lineare" si riferisce al fatto che la relazione tra le variabili viene approssimata all'equazione di una retta, ciò fa capire che ci sono dei prerequisiti che bisogna avere per poter utilizzare questa metodologia:

1. La relazione tra la variabile dipendente Y e la variabile indipendente X deve essere di tipo lineare. Quindi nell'applicazione di questa metodologia si sta facendo una ipotesi molto forte riguardo la relazione lineare tra le variabili.
2. Come detto in precedenza, nella analisi dei dati non si può lavorare direttamente con variabili qualitative ma è necessario che essere vengano trasformate in variabili quantitative e quindi associate a dei numeri. Il secondo prerequisito è quello appunto che le variabili devono essere di tipo quantitativo e quindi associate a dei numeri.
3. Infine, una caratteristica che deve essere presente in ogni modello statistico è che devono esserci un gran numero di osservazioni per poter trarre delle conclusioni che rispecchino a pieno la situazione reale

Ora si vedranno le equazioni per una regressione lineare semplice ed una multipla:

- REGRESSIONE LINEARE SEMPLICE:

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

- REGRESSIONE LINEARE MULTIPLA:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n + \varepsilon$$

Y: come detto in precedenza è la variabile dipendente

β_0: intercetta ovvero il valore che la variabile Y assume quando tutti i regressori sono posti pari a zero

$\beta_1 \ldots \beta_n$: sono i coefficienti dei regressori e ci dicono di quanto varia al variabile di risposta quantitativa a seguito di una variazione unitaria della variabile indipendente

ε: sarebbe l'errore commesso nel calcolo della Y stimata dal nostro modello e quindi di quanto si discosta rispetto al valore reale

Dal grafico (Figura 8) si può osservare l'andamento lineare della funzione di Y, la funzione non parte da 0 ma dal valore assunto dalla intercetta β_0 ed inoltre si può constatare come ad una variazione di X corrisponda una corrispondente variazione in Y ed è proprio questa correlazione che si va a misurare con le analisi di regressione lineare.

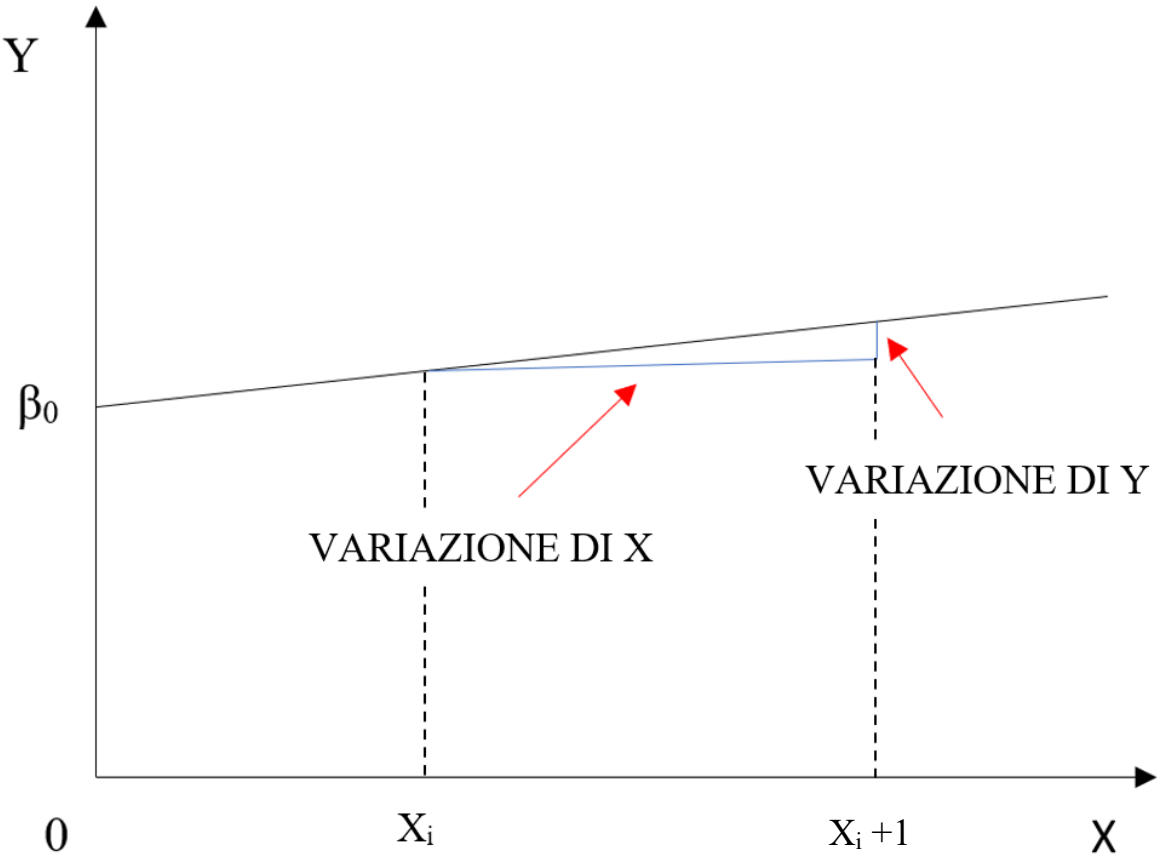

Figura 8: Rappresentazione grafica della retta di regressione

Il termine "correlazione" rimanda al concetto di analisi di correlazione; infatti, la regressione lineare utilizza alcuni concetti di questa analisi ovvero entrambe permettono di ricercare l'effetto che ha una variabile su di un'altra.

Tornando al concetto di errore (ε) esso fa riferimento alla differenza tra il valore reale della variabile dipendente (Y) e quello calcolato con l'applicazione del modello di regressione lineare (Y*).

Si avrà:

$$Y^* = \beta_0 + \beta_1 X$$

quindi:

$$Y = Y^* + \varepsilon$$

e infine si ottiene la formula di ε che è:

$$\varepsilon = Y - Y^*$$

Questo errore (Figura 9) fa riferimento a degli effetti casuali che determinano questa differenza dei due valori di Y ed è la componente stocastica della equazione.

Perché c'è questo errore?
Per 3 motivazioni principali:

1. Come detto in precedenza, una delle ipotesi è che la relazione tra la variabile X e la Y deve essere lineare e nel caso in cui non dovesse essere realmente così questo errore tiene conto della situazione
2. Altra motivazione è che noi abbiamo preso in considerazione delle variabili indipendenti che hanno effetto sulla X e che sono correlate con essa, ma queste potrebbero non essere le uniche ad avere effetto sulla variabile dipendente.
3. Le variabili potrebbero non essere state misurate correttamente.

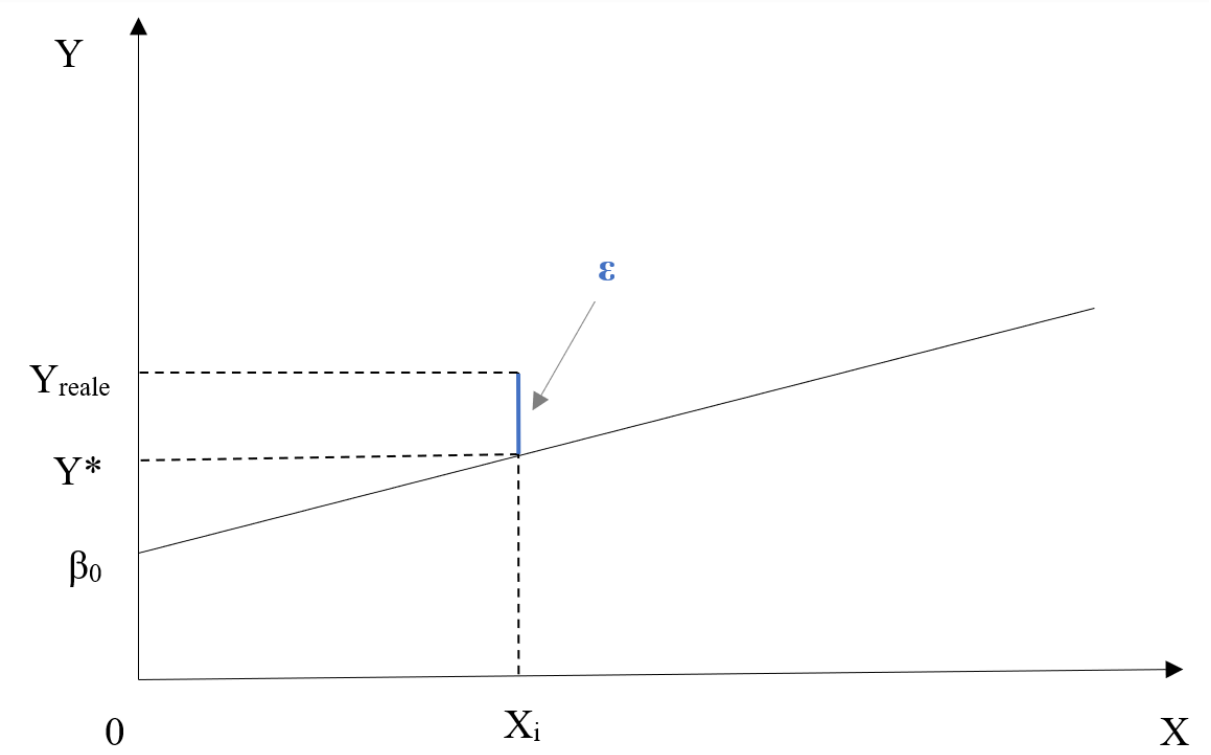

Figura 9: Rappresentazione grafica dell'errore ε

3.2 La regressione lineare applicata al caso studio

Nel caso studio sono state definite le variabili di riferimento per l'applicazione del modello di regressione lineare multipla.

La variabile di risposta quantitativa o variabile dipendente è il PUN, mentre le variabili indipendenti o regressori sono le varie dummy individuate precedentemente che chiameremo $Dummy_i$.

Quindi l'equazione può essere riscritta così:

$$PUN = \beta_0 + \beta_1 Dummy_1 + \cdots + \beta_n Dummy_n + \varepsilon$$

I dati del PUN sono quelli presi dal sito del GME mentre le Dummy sono state calcolate precedentemente ed ognuna di esse tiene conto del mese, del giorno della settimana e dell'ora del giorno in cui è presente quella misurazione di PUN.

L'errore ε è dato dalla differenza tra il PUN reale ed il PUN calcolato con l'equazione della regressione lineare (PUN*), infatti avremo:

$$PUN^* = \beta_0 + \beta_1 Dummy_1 + \cdots + \beta_n Dummy_n$$

Sostituendo:

$$PUN = PUN^* + \varepsilon$$

Quindi:

$$\varepsilon = PUN - PUN^*$$

Presupponendo che le variabili siano state misurate correttamente, questo errore (Figura 10) può essere riferito a due diverse cause:

1. Il fatto che la relazione tra la variabile dipendente PUN e la variabile indipendente Dummy sia stata ipotizzata come lineare
2. Nel PUN noi stiamo tenendo conto solo della stagionalità che ha effetto su esso ma ci potrebbero essere anche altre cause che determinano un aumento o una diminuzione del valore del PUN quali: fattori politici, fattori economici o altri fattori.

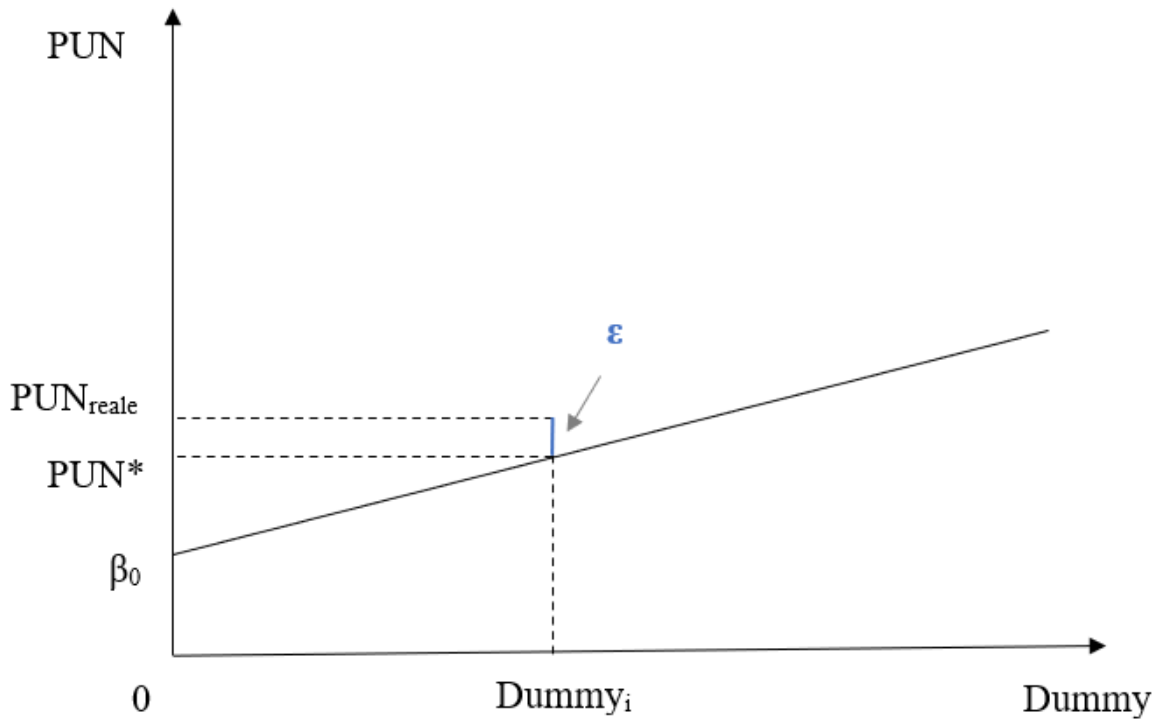

Figura 10: Rappresentazione dell'errore ε applicato al caso studio

Per fare questa analisi bisognerà creare un vettore per la variabile dipendente PUN ed una serie di vettori per le variabili indipendenti in base a quanto sono le colonne delle dummy individuate precedentemente.

Si può passare su Python alla creazione dei vettori:

X = data.drop(['Anno', 'Mese', 'Giorno_Mese', 'Giorno_Settimana', 'Ora', 'PUN', 'Comb'], axis=1).values

Innanzitutto, si lavora sul Dataframe, creato in precedenza, chiamato “data”, quello che si sta facendo è escludere alcune colonne del dataset per fare in modo che rimangano solo quelle utili per l’analisi.

Qui si stanno escludendo dal dataset le colonne: “Anno”, “Mese”, “Giorno_Mese”, “Giorno_Settimana”, “Ora”, “PUN”, “Comb”.

In questo modo le uniche colonne che verranno incluse nel vettore relativo ai regressori sono le colonne legate alle variabili dummy.

È stato scelto di procedere in questo modo perché era quello più rapido in quanto è presente un considerevole numero (più di 2000) di colonne legate alle dummy.

Tutto ciò è stato fatto grazie all'utilizzo della funzione "drop" con la quale, avendo specificato axis=1 (axis =1 è il riferimento per le colonne), si rimuovono le colonne specificate.

Se si osserva bene si nota un ".values" alla fine, questo ci permette di rappresentare questi elementi presi dal Dataframe come un vettore (array) di NumPy utile per effettuare delle operazioni successive.

Ora si può passare alla creazione del vettore relativo alla variabile di risposta quantitativa Y:

Y = data['PUN'].values

In questo caso si specifica una colonna del DataFrame "data", ovvero la colonna "PUN", che rappresenterà la nostra variabile dipendente.

Alla fine, è sempre la funzione ".values" che, come in precedenza, permette la rappresentazione sottoforma di vettore di NumPy.

3.3 Il metodo dei minimi quadrati

Nell'ambito della regressione lineare esistono diversi metodi per effettuare questa analisi, uno di questi, che è anche quello che verrà utilizzato nello studio, è il metodo dei minimi quadrati (in inglese *Least Square Method*).

Questo è uno dei metodi statistici più utilizzati nell'analisi dati e per la creazione dei modelli predittivi.

Sostanzialmente consiste nel calcolo dei coefficienti:

- β_0 ovvero il valore assunto dalla variabile dipendente Y quando tutti i regressori sono posti uguali a zero.
- $\beta_1 \dots \beta_n$ (con n numero totale di valori) ovvero i coefficienti legati ai regressori che tengono traccia della informazione relativa alla correlazione tra il regressore associato e la variabile di risposta quantitativa.

Ricordiamo l'errore ε definito come la differenza tra il valore reale della variabile dipendente Y e quello calcolato Y*:

$$\varepsilon_i = Y_i - Y_i^*$$

E dove Y* è pari a:

$$Y_i^* = \beta_0 + \beta_i X_i$$

Questo errore è alla base della analisi dei minimi quadrati; infatti, si calcolano i valori dei vari β minimizzando la somma dei quadrati degli errori ε (da qui il nome “minimi quadrati”).

Si tratterà di minimizzare la seguente espressione:

$$\sum_{i=1}^{n}(Y_i - Y_i^*)^2$$

Sostituendo l’espressione di Y* diventa:

$$\sum_{i=1}^{n}(Y_i - \beta_0 - \beta_i X_i)^2$$

Perché si fa il quadrato degli errori e non si considerano direttamente gli errori stessi senza elevarli al quadrato?

Questo deriva da due ragioni fondamentali:

1. La prima ragione deriva dal fatto che, se i valori non venissero elevati al quadrato, ci si ritroverebbe in una situazione in cui valori positivi e valori negativi si sommerebbero tra di loro, portando ad una errata valutazione. Quindi il quadrato fa in modo che tutti abbiano lo stesso segno e possano essere sommati senza che ci siano delle alterazioni dovute al diverso segno.
2. La seconda ragione è che elevando al quadrato gli errori con un valore più alto pesano maggiormente e così gli si dà più risalto.

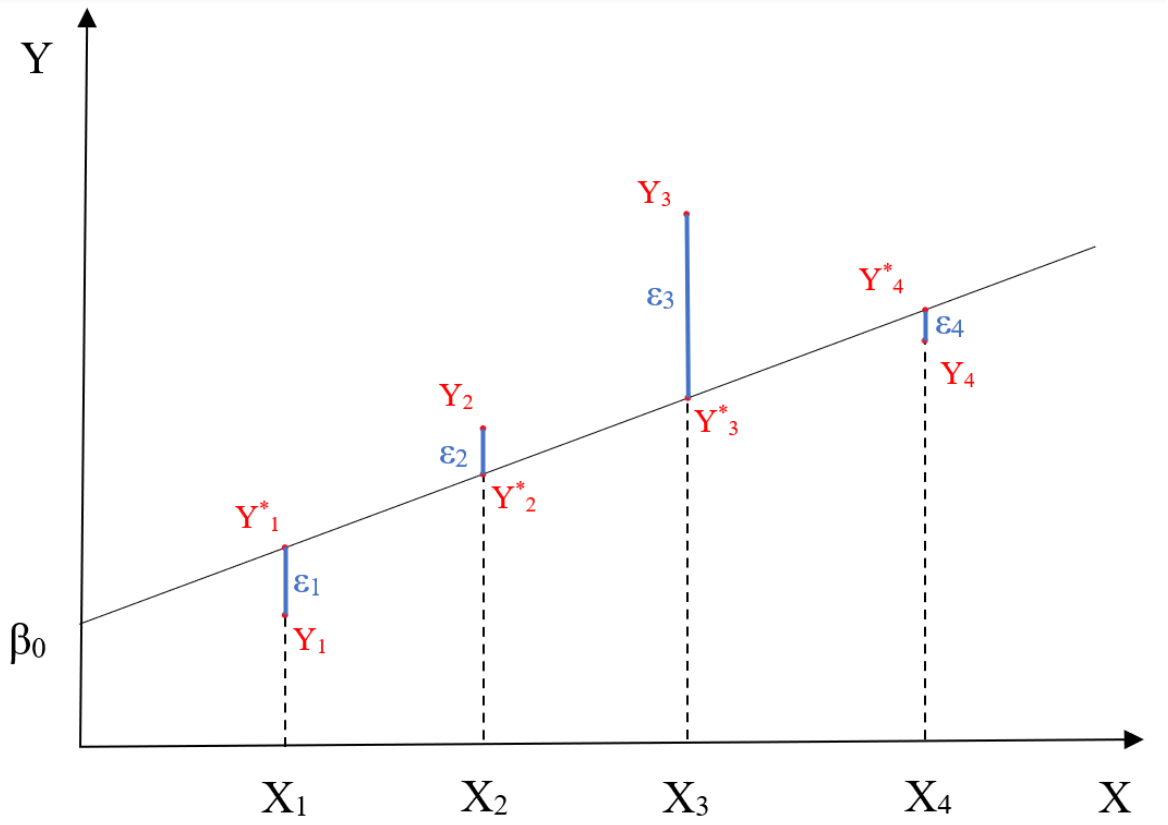

Figura 11: Rappresentazione degli errori ε in una analisi di regressione multipla

Ci troviamo in una situazione in cui siamo in presenza di un numero di errori pari al numero di variabili di risposta quantitativa Y presenti (Figura 11); perciò, più valori avremo a disposizione più la misurazione sarà accurata ed il metodo potrà fornire dei risultati soddisfacenti.

Una cosa importante da tenere in conto quando si vuole effettuare questa analisi è vedere se ci sono possibili correlazioni tra le variabili ed eventualmente risolverle eliminando una delle due variabili coinvolte; ciò perché si avrebbe il problema di avere una matrice detta "singolare" ovvero con la presenza di multicollinearità e non si potrebbe applicare il metodo dei minimi quadrati correttamente.

Dato che questo metodo utilizza il quadrato degli errori, lo rende molto sensibile a valori un po' fuori scala, quindi, bisogna stare attenti ad impostare al meglio il dataset prima dell'analisi per cercare quanto più possibile di limitare questa evenienza.

I parametri relativi alla equazione di regressione lineare possono essere rappresentati come vettori e quindi si può utilizzare la notazione matriciale:

$$Y = \begin{bmatrix} Y_1 \\ \vdots \\ \vdots \\ Y_n \end{bmatrix} \qquad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{bmatrix} \qquad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Per il vettore della X c'è una prima colonna identità ed è riferito alla intercetta presente nella equazione di regressione lineare:

$$X = \begin{bmatrix} 1 & X_{11} & \dots & X_{1m} \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ 1 & X_{n1} & \dots & X_{nm} \end{bmatrix}$$

Con il metodo dei minimi quadrati si possono calcolare le X avendo a disposizione i valori delle Y reali e quindi vedere la correlazione tra la variabile indipendente X e la variabile dipendente Y.

Ora si vedrà come effettuare questo calcolo con il metodo matriciale, trascurando il contributo dell'intercetta β_0.

L'equazione si riscrive così:

$$Y = \beta X$$

Dove la β è la nostra incognita.

Si applica il metodo dei minimi quadrati al caso matriciale ed ipotizziamo di avere solo 4 misurazioni per semplicità e 2 coefficienti X:

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} \qquad X = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \\ X_{31} & X_{32} \\ X_{41} & X_{42} \end{bmatrix}$$

Ora si fa la trasposta della matrice X:

$$X^T = \begin{bmatrix} X_{11} & X_{21} & X_{31} & X_{41} \\ X_{12} & X_{22} & X_{32} & X_{42} \end{bmatrix}$$

Si fa il prodotto scalare tra la matrice trasposta X^T e il vettore X:

$$X^T \times X = \begin{bmatrix} X_{11} & X_{21} & X_{31} & X_{41} \\ X_{12} & X_{22} & X_{32} & X_{42} \end{bmatrix} \times \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \\ X_{31} & X_{32} \\ X_{41} & X_{42} \end{bmatrix}$$

Successivamente il prodotto scalare tra la matrice trasposta X^T e il vettore Y:

$$X^T \times Y = \begin{bmatrix} X_{11} & X_{21} & X_{31} & X_{41} \\ X_{12} & X_{22} & X_{32} & X_{42} \end{bmatrix} \times \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix}$$

Fatto ciò, si riscrive l'equazione e si può calcolare la β:

$$(X^T \times X)\beta = (X^T \times Y)$$

3.4 Applicazione del metodo dei minimi quadrati al caso studio

È stato applicato il metodo matriciale su Python e i vettori relativi alla variabile di risposta quantitativa Y e dei regressori X sono rispettivamente:

$$PUN = \begin{bmatrix} PUN_{01\_01\_2005} \\ \vdots \\ \vdots \\ PUN_{31\_12\_2010} \end{bmatrix}$$

Il PUN è la variabile dipendente, mentre:

$$Dummy = \begin{bmatrix} Dummy_{Ora1\_Lunedì\_Gennaio\_01\_01\_2005} & \cdots & \cdots & Dummy_{Ora24\_Domenica\_Dicembre\_01\_01\_2005} \\ \vdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \vdots \\ Dummy_{Ora1\_Lunedì\_Gennaio\_31\_12\_2010} & \cdots & \cdots & Dummy_{Ora24\_Domenica\_Dicembre\_31\_12\_2010} \end{bmatrix}$$

Le variabili dummy sono le variabili indipendenti.

Adesso si andrà ad analizzare il codice utilizzato per l'analisi dei minimi quadrati su Python.

Si inizia facendo la trasposta della matrice X, quella relativa alle variabili dummy, invertendo le righe con le colonne:

X_T = X.transpose()

Si esegue il prodotto matriciale tra la matrice trasposta "X_T" e il vettore X (dummy) e si salvano i risultati in una nuova variabile "X_X":

X_X = np.dot(X_T, X)

Successivamente il prodotto matriciale tra la matrice trasposta "X_T" e il vettore Y (PUN) e si salvano i risultati in una nuova variabile "X_Y":

X_Y = np.dot(X_T, Y)

Si possono calcolare i coefficienti risolvendo il sistema con le equazioni lineari rappresentate dai termini "X_X" e "X_Y", sfruttando la funzione "solve":

coefficients = np.linalg.solve(X_X, X_Y)

Ora si inseriscono questi coefficienti in un DataFrame chiamato "coeff_data" in cui è presente una sola colonna chiamata "Coefficient" e sulle righe ci sono i valori dei coefficienti calcolati col metodo dei minimi quadrati:

coeff_data = pd.DataFrame({'Coefficient': coefficients})

In definitiva, quello che è stato fatto è creare un una colonna chiamata "PUN_stag" i cui valori sono il prodotto di ogni elemento della riga con il coefficiente associato, quindi in questo caso sarebbe il prodotto dei coefficienti delle dummy con le dummy di quell'ora e successivamente tutti questi valori vengono sommati per dar vita al valore finale del PUN calcolato.

L'intenzione è quella di inserire affiancati i valori del PUN reale e di quello calcolato con questo metodo così che sia più facile la creazione di grafici e il confronto.

4. Risultati del modello

4.1 Analisi dei risultati e grafici di confronto

Prima di poter analizzare e discutere i risultati del modello bisogna manipolare il dataset per poter effettuare un controllo ottimale:

confronto = data.drop(['Anno', 'Mese', 'Giorno_Mese', 'Giorno_Settimana', 'Ora', 'Comb'], axis=1)

Si crea un nuovo DataFrame chiamato “confronto” che contiene tutte le colonne presenti in “data” escludendo: “Anno”, “Mese”, “Giorno_Mese”, “Giorno_Settimana”, “Ora”, “Comb”.

confronto = confronto[[col for col in confronto.columns if col != 'PUN'] + ['PUN']]

La colonna con i dati del PUN viene inserita alla fine del DataFrame “confronto” e questo ci sarà utile in seguito.

riga_vuota = {col: None for col in confronto.columns}

Così si crea un nuovo DataFrame con una riga vuota.

confronto = pd.concat([pd.DataFrame([riga_vuota]), confronto], ignore_index=True)

Si aggiunge la riga vuota al dataset “confronto” facendo in modo che essa sia la prima riga presente.

confronto.iloc[0, :len(coefficients)] = coefficients

La riga vuota viene riempita con i valori dei coefficienti calcolati con il metodo dei minimi quadrati (i vari β).

confronto['PUN_stag'] = None

Una colonna dal nome “PUN_stag” viene creata e ad essa vengono assegnati valori nulli (“None”).

Bisogna calcolare i valori chiamati “PUN_stag” che sarebbero i valori della variabile indipendenti calcolati con l’equazione di regressione lineare e quindi si fa il calcolo della somma pesata per ottenere tali valori:

```
colonne_numeriche = confronto.columns.difference(['PUN', 'PUN_stag'])
```

```
confronto['PUN_stag'] = confronto.apply(lambda row: np.sum(row[colonne_numeriche] * confronto.loc[0, colonne_numeriche]), axis=1)
```

```
data_anno = data[['Anno', 'Mese', 'Giorno_Mese', 'Giorno_Settimana', 'Ora']].copy()
```

Quest’ultima funzione crea un nuovo dataset chiamato “data_anno” in cui vengono inserite le colonne indicate copiandole dal DataFrame “data”.

```
riga_vuota = pd.DataFrame([{}], columns=data_anno.columns)
```

```
data_anno = pd.concat([riga_vuota, data_anno], ignore_index=True)
```

Servono per creare una riga vuota ed inserirla in cima al DataFrame “data_anno”.

```
risultato_finale = pd.concat([data_anno, confronto], axis=1)
```

Infine, si uniscono “data_anno” e “confronto” ottenendo così il dataset finale, chiamato “risultato_finale”, su cui andremo a creare dei grafici per verificare l’accuratezza del modello.

Finito di sistemare la disposizione delle colonne e dei dati si può passare alla fase di verifica del modello e per fare ciò è stato inserito un codice in cui, modificando i valori di “Anno”, “Mese”, “Giorno_Mese” ed “Ora”, si può selezionare un periodo in cui effettuare il confronto.

```
filtered_data = risultato_finale[
    (risultato_finale['Anno'] == 2008) &
    (risultato_finale['Mese'] == 'Novembre') &
    (risultato_finale['Giorno_Mese'].between(1, 30)) &
    (risultato_finale['Ora'] == 12)]
```

La fase di creazione dei grafici per un confronto visivo viene realizzata sfruttando la libreria di matplotlib che permette, appunto, la creazione di grafici di alta qualità.

Prima di tutto creiamo la struttura del grafico:

plt.figure(figsize=(10, 6))

Ora si può passare alla creazione dei grafici del "PUN" e "PUN_stag" che verranno inseriti in una stessa figura così da poter essere confrontati più facilmente.

Creazione grafico PUN:

plt.plot(filtered_data['PUN'], label='PUN', marker='o')

e creazione grafico "PUN_stag":

plt.plot(filtered_data['PUN_stag'], label='PUN_stag', marker='o')

Infine, aggiungiamo titoli e legende ai grafici:

plt.title('Confronto tra PUN e PUN_stag')
plt.xlabel('Indice')
plt.ylabel('Valore')
plt.legend()

Prima di andare a visualizzare i risultati nello specifico, è utile spiegare cosa è stato fatto in questo studio sfruttando Python.

Quello che è stato fatto, sostanzialmente, è la creazione di un modello predittivo ottenuto attraverso l'applicazione di un modello di regressione lineare, in questo caso quello dei minimi quadrati, che sfrutta l'informazione contenuta nella dummy associata ad un'ora del giorno, un giorno della settimana ed un mese ben precisi per calcolare la correlazione presente tra il PUN reale e la stagionalità (ovvero il momento dell'anno in cui ci si trova).

Ciò che si vuole ottenere è replicare il pattern del PUN mostrando come l'analisi di stagionalità possa essere utile per prevedere un po' cosa accadrà al prezzo dell'energia elettrica in un dato periodo dell'anno.

In definitiva quello che si vedrà in questi grafici è la differenza tra il valore del PUN reale e di quello calcolato con la stagionalità, questo è esattamente l'errore ε di cui si è parlato molto in questo elaborato.

Si può procedere con la visualizzazione dei grafici per diversi periodi ed ore del giorno, in particolare si sceglie un mese per ogni stagione:

- INVERNO → gennaio
- PRIMAVERA → maggio
- ESTATE → agosto
- AUTUNNO → novembre

Sono stati scelti i giorni dal 20 al 30 di questi mesi per ottenere una visualizzazione più chiara dell'andamento ed inoltre il periodo su cui verranno creati questi grafici è 2006 al 2008.

Iniziamo dall'inverno ed in particolare dal 20 al 30 gennaio (Figura 12, 13 e 14).

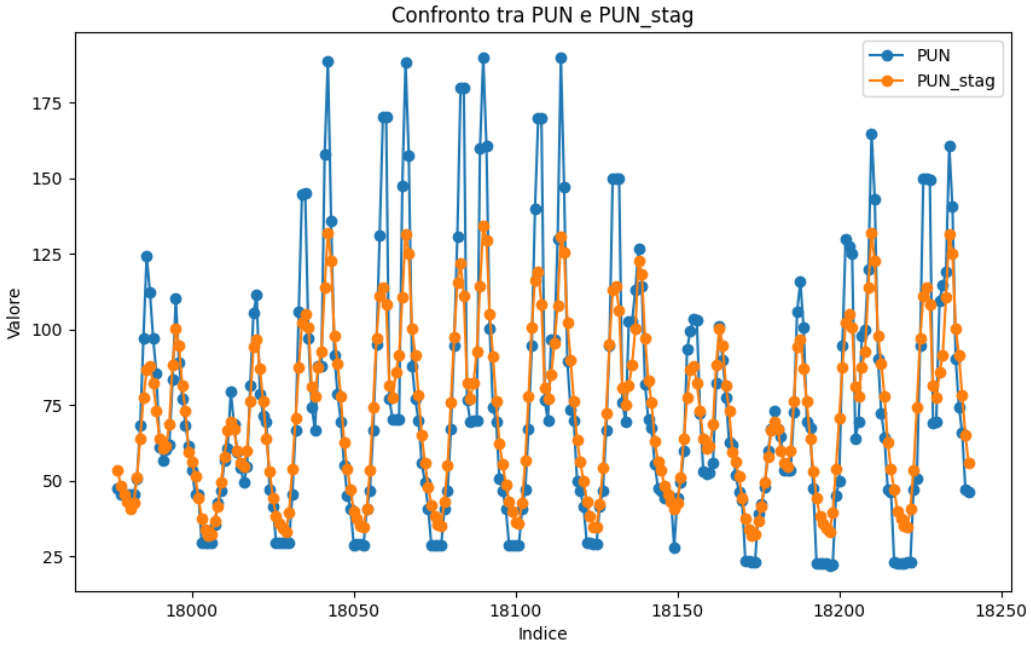

Figura 12: Confronto fra PUN reale e PUN calcolato con la stagionalità nel periodo dal 20 al 30 gennaio 2006

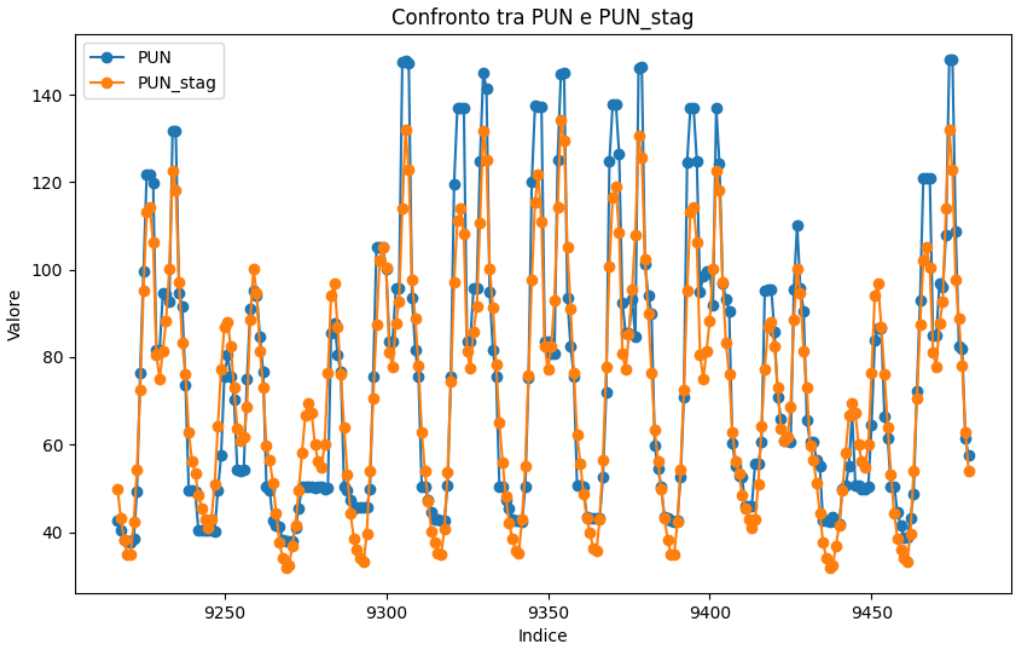

Figura 13: Confronto fra PUN reale e PUN calcolato con la stagionalità nel periodo dal 20 al 30 gennaio 2007

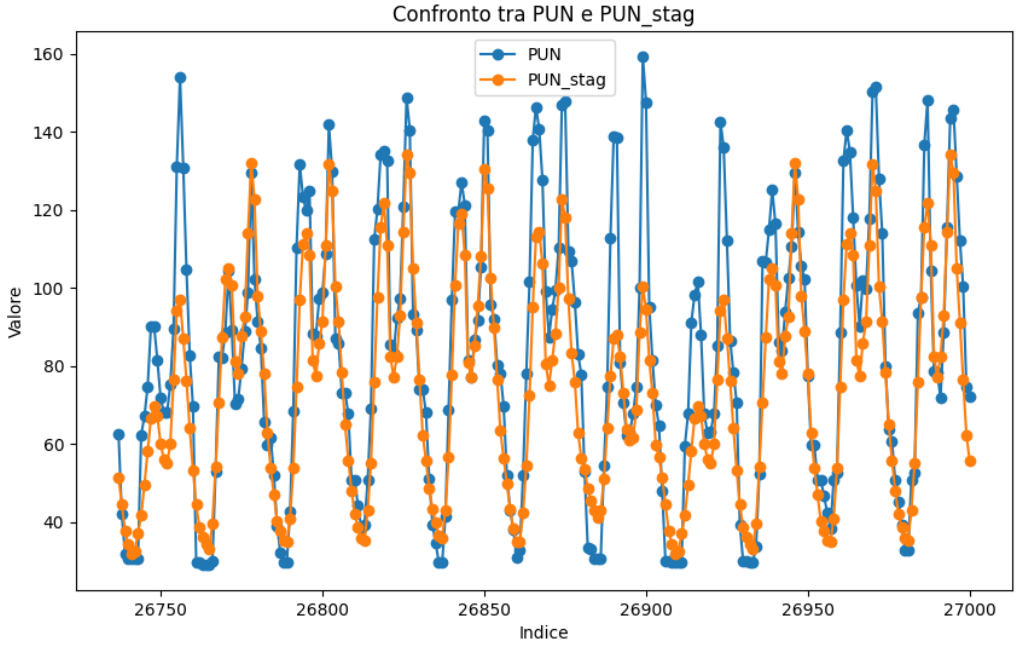

Figura 14: Confronto fra PUN reale e PUN calcolato con la stagionalità nel periodo dal 20 al 30 gennaio 2008

Si può vedere che per l'inverno ed in questo caso, i giorni dal 20 al 30 gennaio, il modello riesce a seguire l'andamento del PUN reale.

Si passa alla primavera con i giorni che vanno dal 20 al 30 maggio (Figura 15, 16 e 17).

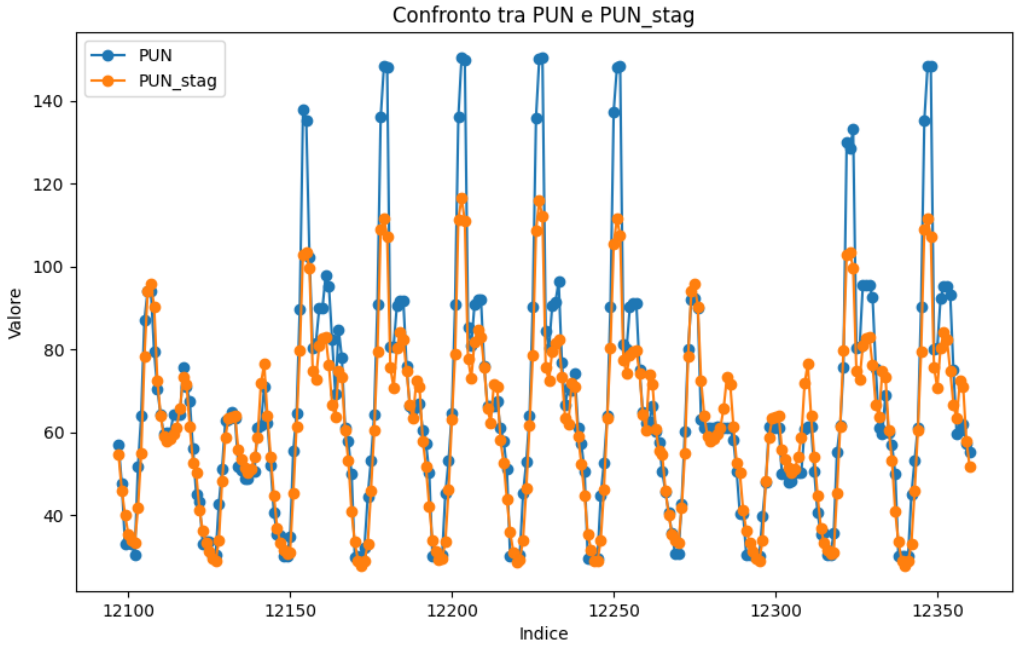

Figura 15: Confronto fra PUN reale e PUN calcolato con la stagionalità nel periodo dal 20 al 30 maggio 2006

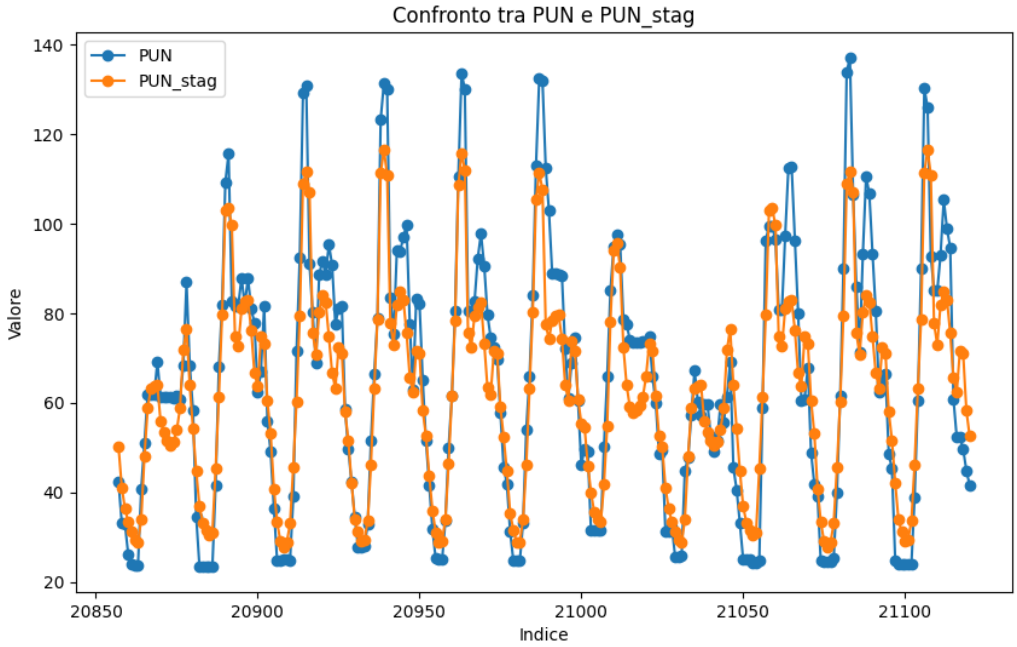

Figura 16: Confronto fra PUN reale e PUN calcolato con la stagionalità nel periodo dal 20 al 30 maggio 2007

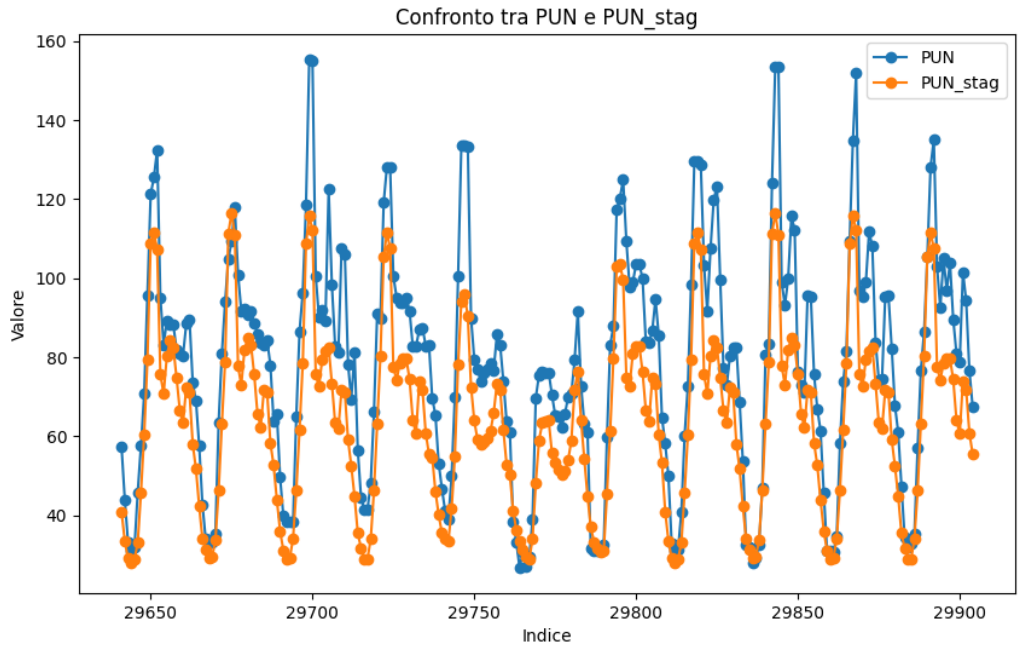

Figura 17: Confronto fra PUN reale e PUN calcolato con la stagionalità nel periodo dal 20 al 30 maggio 2008

In questo secondo caso anche riusciamo a replicare l'andamento, solo per maggio 2008 ci sono dei valori del PUN reali più alti del previsto, ciò potrebbe essere dovuto a numerosi fattori, tra cui può aver inciso la crisi finanziaria del 2008.

L'estate con i giorni dal 20 al 30 agosto (Figura 18, 19 e 20).

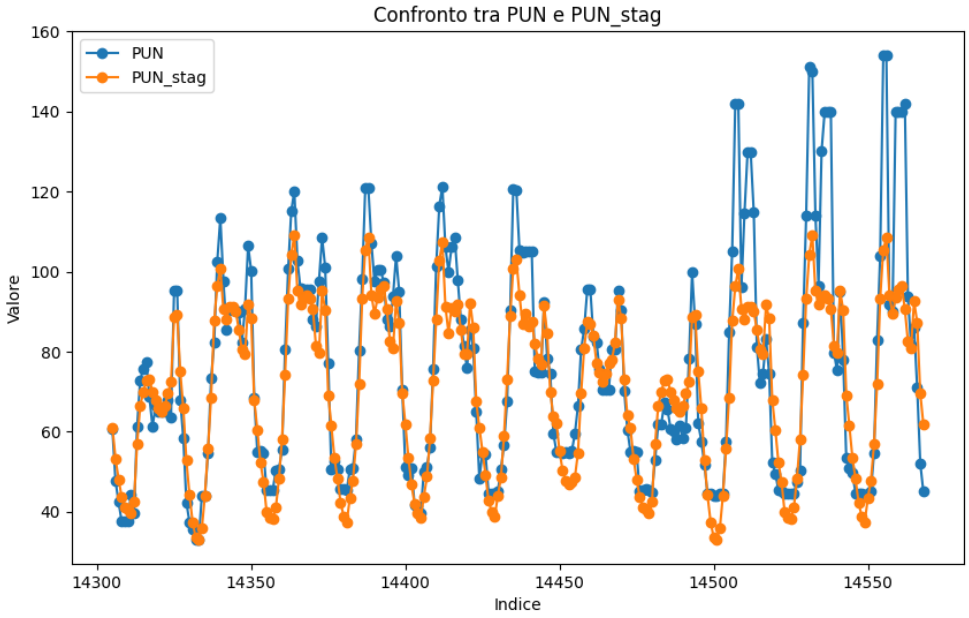

Figura 18: Confronto fra PUN reale e PUN calcolato con la stagionalità nel periodo dal 20 al 30 agosto 2006

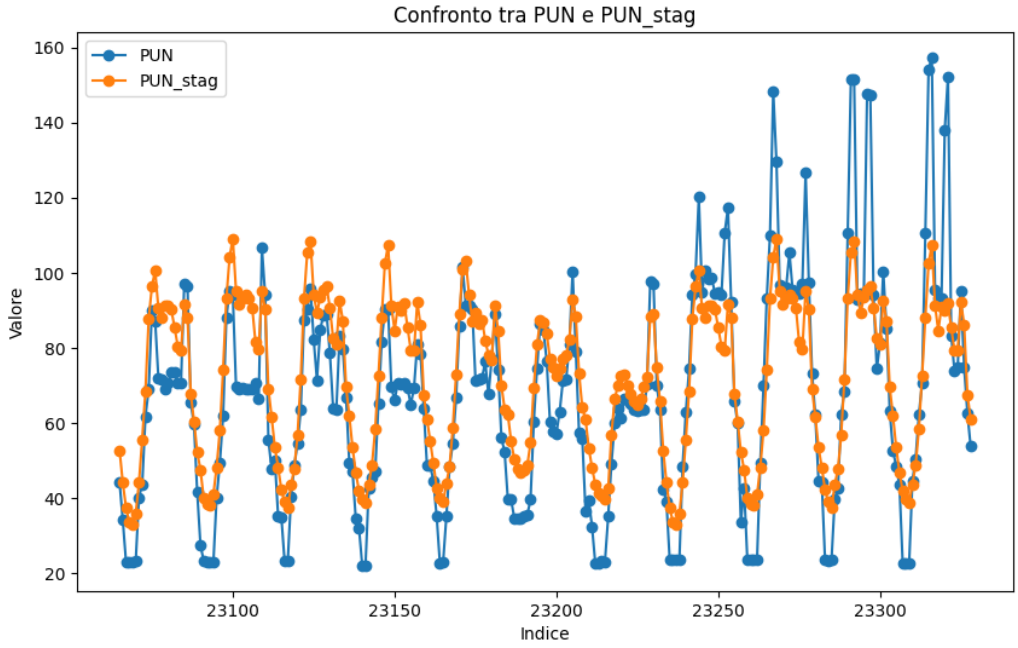

Figura 19: Confronto fra PUN reale e PUN calcolato con la stagionalità nel periodo dal 20 al 30 maggio 2007

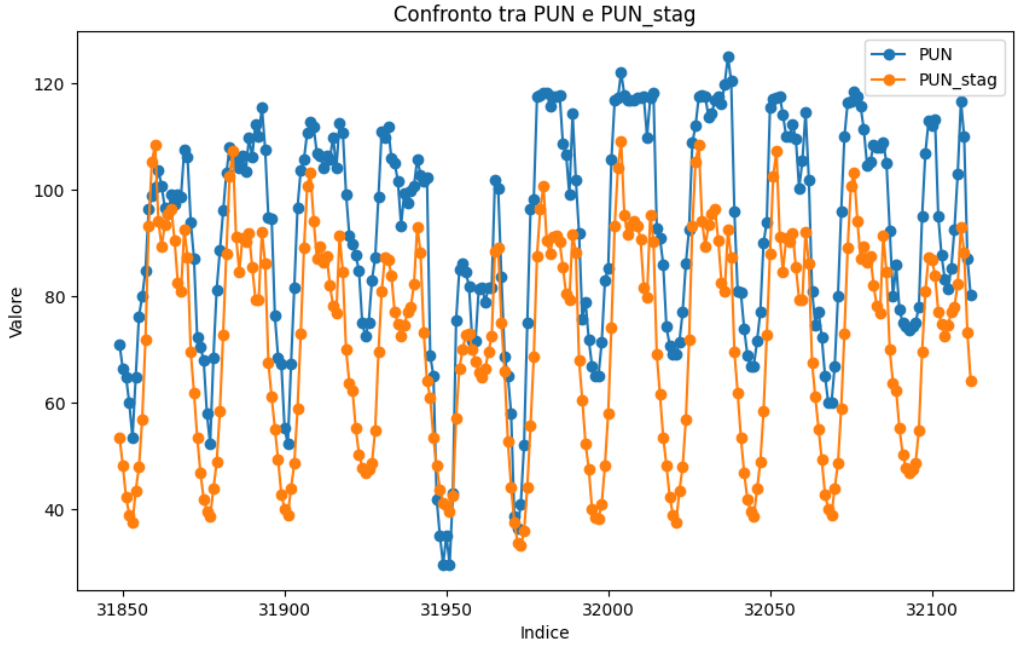

Figura 20: Confronto fra PUN reale e PUN calcolato con la stagionalità nel periodo dal 20 al 30 maggio 2008

Per agosto si può notare il fatto che i valori del PUN variano di tanto e le ragioni potrebbero essere innumerevoli ma, in ogni caso, il modello riesce a seguire il trend di prezzo.

L’autunno con i giorni dal 20 al 30 novembre (Figura 21, 22 e 23).

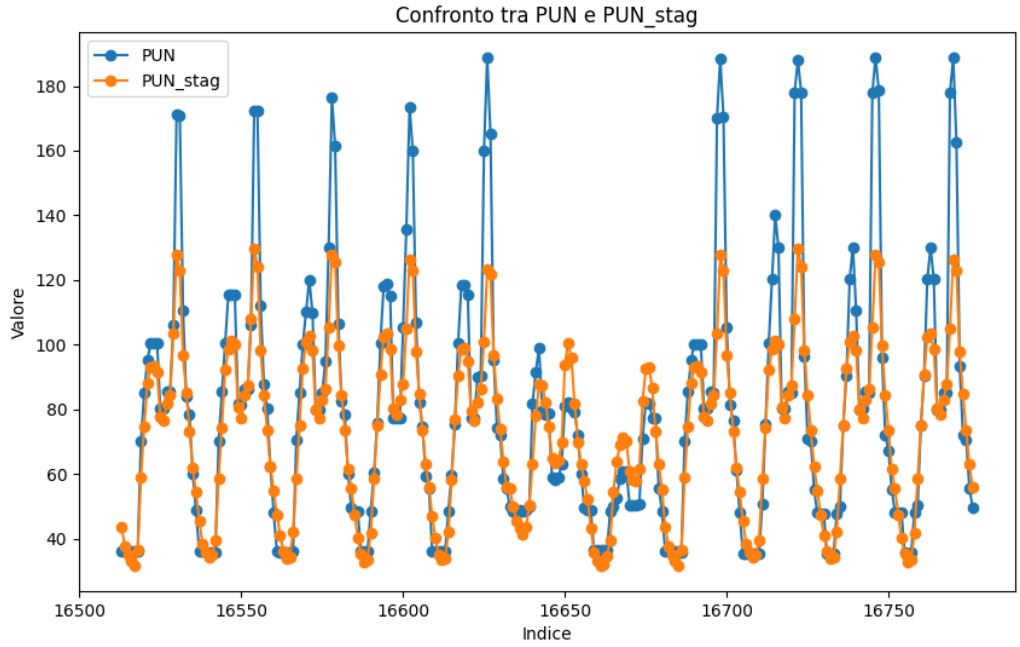

Figura 21: Confronto fra PUN reale e PUN calcolato con la stagionalità nel periodo dal 20 al 30 novembre 2006

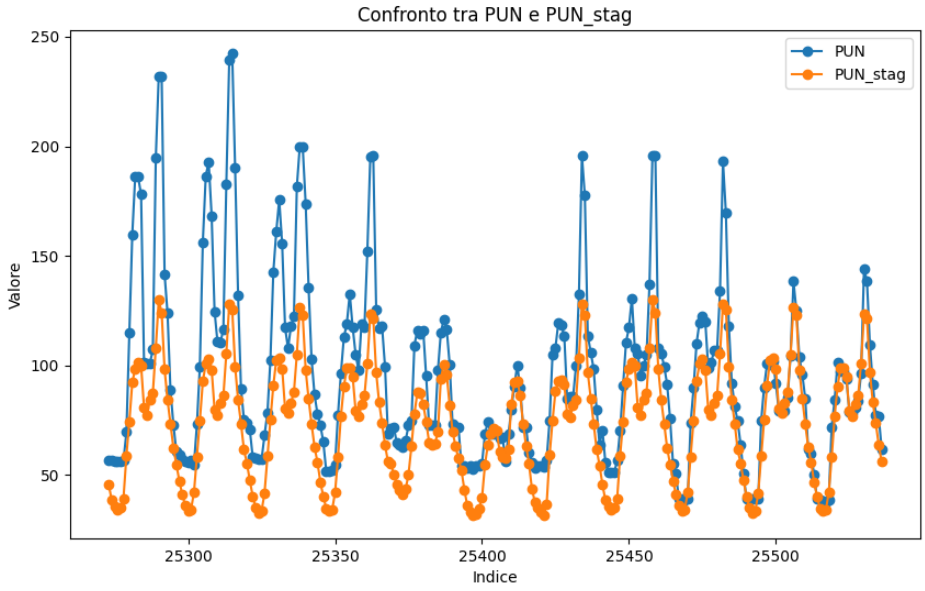

Figura 22: Confronto fra PUN reale e PUN calcolato con la stagionalità nel periodo dal 20 al 30 maggio 2007

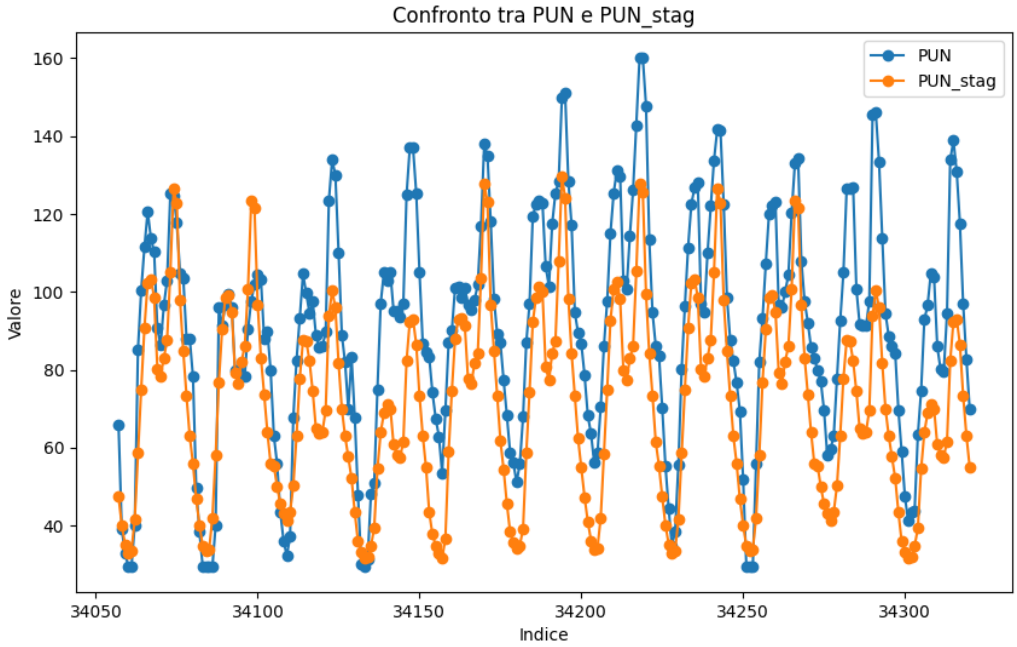

Figura 23: Confronto fra PUN reale e PUN calcolato con la stagionalità nel periodo dal 20 al 30 maggio 2008

Nel caso di novembre si osserva come ci siano dei valori particolarmente elevati del PUN, addirittura nel 2007 si raggiungono valori vicini ai 250 €/MWh.

Questo fa capire come il tema della previsione del prezzo elettrico sia un argomento estremamente complesso e l'aver calcolato il pattern di stagionalità è sicuramente un primo passo per poter effettuare delle analisi aggiuntive.

4.2 Ultimo appunto sull'errore prodotto dal modello

Come già detto in precedenza l'errore quadratico è dato dalla seguente formula:

$$\sum_{i=1}^{n} (PUN_i - PUNstag_i)^2$$

n: è il numero di valori totali, in questo caso, considerando i valori dal primo gennaio 2005 al 31 dicembre 2010, si hanno 52584 valori

Si può calcolare l'errore quadratico medio semplicemente dividendo per n, quindi:

$$\varepsilon_{medio}^2 = \frac{\sum_{i=1}^{n}(Y_i - \beta_0 - \beta_i X_i)^2}{n}$$

Si ottiene un valore di errore quadratico medio pari a 23,823 €/MWh che paragonato con il valore medio del PUN reale che è di 69,87 €/MWh è un errore pari al 34%.

Questo errore pari al 34 % è dovuto al fatto che la stagionalità non è l'unico elemento che influisce sul prezzo dell'energia elettrica.

Se si vuole ridurre questo errore bisognerebbe includere ulteriori repressori come: il mix energetico, la domanda di energia elettrica, il prezzo del gas ed altri; in questo studio ci si è concentrati sulla stagionalità e del suo effetto sul PUN.

5. Conclusioni

In questo elaborato sono stati trattati diversi temi ed utilizzati diversi strumenti come, ad esempio, l'utilizzo di Python per le varie analisi.

Python si è dimostrato uno strumento molto versatile ed utile per la realizzazione del modello permettendo modifiche ed aggiunte in termini brevi.

Sicuramente il prezzo dell'energia elettrica PUN e la sua determinazione è un argomento molto complesso e sul quale hanno effetto molti fattori.

La stagionalità sicuramente è uno di questi e si è notato un certo trend di stagionalità ma chiaramente ci sono anche altri elementi, basti pensare alla crisi del 2008 per esempio che sicuramente ha avuto un impatto su tutti i mercati compreso il mercato dell'approvvigionamento elettrico.

Per calcolare la stagionalità si sono dimostrate estremamente utili le variabili dummy che permettono di analizzare dei dati qualitativi come, ad esempio, un giorno della settimana e tramite di esse si ottengono i parametri numerici utili per l'analisi dei minimi quadrati.

Quest'ultima, dopo un periodo di comprensione su come applicare al meglio il metodo, ha permesso il calcolo dei vari coefficienti legati alla stagionalità e quindi ad un momento preciso dell'anno.

I coefficienti ci hanno permesso di calcolare il PUN previsto dal modello di stagionalità ed effettuare il confronto con quello reale, permettendoci di verificare che funziona piuttosto bene e sarà sicuramente utile per analisi future integrando, ad esempio, prezzi del gas e mix energetico

Oltre a questi due fattori che dovranno essere tenuti in considerazione, a mio avviso, dovrà essere incluso il mercato del bilanciamento in quanto esso assumerà sempre di più un ruolo chiave.

In definitiva la stagionalità è solo un primo passo verso la creazione di un modello completo che permetta di calcolare il prezzo dell'energia elettrica ma di sicuro pone le basi per poter effettuare ulteriori analisi.

Bibliografia

Abdi, H. (n.d.). *The Method of Least Squares*. University of Texas at Dallas. https://personal.utdallas.edu/~herve/Abdi-LeastSquares06-pretty.pdf

Author: Ricky Dunbar Solar Energy Engineer, Engineer, R. D. S. E., Psychometrician, E. R., & LearnDataSci, E. B. F. of. (n.d.). *Python numpy tutorial: An applied introduction for beginners*. Learn Data Science - Tutorials, Books, Courses, and More. https://www.learndatasci.com/tutorials/applied-introduction-to-numpy-python-tutorial/

Billé, A. G., Gianfreda, A., Del Grosso, F., & Ravazzolo, F. (2023). Forecasting electricity prices with expert, linear, and nonlinear models. *International Journal of Forecasting*, *39*(2), 570–586. https://doi.org/10.1016/j.ijforecast.2022.01.003

Bonnini, S., & Ragazzi, S. (n.d.). *Capitolo 12: La regressione lineare semplice*. Facoltà di Economia, Università di Ferrara. http://math.unife.it/informatica/insegnamenti/statistica-applicata/materiale/levine-capitolo-12.pdf

Campobasso, F. (n.d.). Variabili qualitative e dummies nel modello di regressione lineare (modifica dell'intercetta e della pendenza). https://www.uniba.it/docenti/campobasso-francesco/attivita-didattica/Levariabilidummiesnellaregressionelineare.ppt

Capiluppi, C. (n.d.). *La regressione lineare*. Facoltà di Scienze della Formazione, Università degli Studi di Verona. https://www.dsu.univr.it/documenti/OccorrenzaIns/matdid/matdid324170.pdf

Capitolo 11: I minimi quadrati. Università del Salento. (n.d.). http://www.dmf.unisalento.it/LaureeScientificheFisica/Download/fisicamoderna/MinimiQuadrati.pdf

Capparelli, S. (n.d.). *0.1 Metodo dei Minimi Quadrati*. Dipartimento di Scienze di Base e Applicate per l'Ingegneria, Università degli Studi di Roma "La Sapienza." https://www.sbai.uniroma1.it/~stefano.capparelli/didattica/Geometria%2014-15/minimi%20quadrati.pdf

Cos'è python: Linguaggio di Programmazione Python. Cos'è Python | Linguaggio di programmazione Python. (n.d.). https://www.python.it/about/

ENERGIA E CLIMA IN ITALIA: RAPPORTO TRIMESTRALE. Gestore dei Servizi Energetici (GSE). (2022, November). https://www.gse.it/documenti_site/Documenti%20GSE/Rapporti%20statistici/GSE%20Relazione%20trimestrale.pdf

Energia, V. (2023, September 11). *Il sistema di Prezzo marginale dell'energia*. Solar Cash. https://www.solarcash.eu/blog/il-sistema-di-prezzo-marginale-dellenergia/

Garden, T. (2021, April 23). *Analisi dei Dati Con Python: Un approccio graduale*. Talent Garden. https://blog.talentgarden.com/it/blog/data/analisi-dei-dati-python-approccio-passo-dopo-passo

Gianfreda, A., Parisio, L., & Pelagatti, M. (2018). A review of balancing costs in Italy before and after res introduction. *Renewable and Sustainable Energy Reviews*, *91*, 549–563. https://doi.org/10.1016/j.rser.2018.04.009

Il Green deal Europeo. Commissione europea. (n.d.). https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/european-green-deal_it

Il metodo dei minimi quadrati. Istituto Nazionale di Fisica Nucleare, Sezione di Padova. (n.d.). https://userswww.pd.infn.it/~montag/didattica/ottica/errori_04.pdf

King, S. (2022). *What is Python?*. Amazon. https://aws.amazon.com/it/what-is/python/

La regressione lineare (semplice). Università degli Studi Mediterranea di Reggio Calabria. (n.d.). https://www.unirc.it/documentazione/materiale_didattico/1465_2017_437_28908.pdf

Least squares method. (n.d.-a). https://arturo.imati.cnr.it/marini/metnum/Ch.5-least-squares-notes.pdf

Least squares method. Università degli Studi di Pavia. (2019, November 17). https://www-dimat.unipv.it/sangalli/numerical_methods_eng_sciences/5%20-%20Least%20Squares.pdf

Levine, D. M., Krehbiel, T. C., Berenson, M. L., & Piccarreta, R. (2002). *Statistica*. Apogeo.

Massari, R. (2016). *Il modello di regressione lineare*. Università degli Studi di Roma "La Sapienza." http://www.diss.uniroma1.it/moodle2/pluginfile.php/6816/mod_resource/content/1/MASSARI%20linear-regression.pdf

Mercati Elettrici. GME - i mercati - Mercato Elettrico - MGP, MI, MPEG, MSD. (n.d.). https://www.mercatoelettrico.org/it/mercati/mercatoelettrico/mpe.aspx

Metodo dei Minimi Quadrati. Università degli Studi di Napoli Federico II. (n.d.). https://www.docenti.unina.it/webdocenti-be/allegati/materiale-didattico/460941

Miami University. (n.d.-b). Chapter 7, Dummy Variable. https://www.fsb.miamioh.edu/lij14/311_2014_0416.pdf

Minimi quadrati, regressione lineare. Università degli Studi dell'Aquila. (n.d.). https://people.disim.univaq.it/~serva/teaching/previous/lezione11-12.pdf

Pandas Package overview. Package overview - pandas 2.1.2 documentation. (n.d.). https://pandas.pydata.org/docs/getting_started/overview.html

Pandas. Libreria Pandas. (n.d.). https://disi.unitn.it/~passerini/teaching/2020-2021/informatica/slides/32_pandas/pandas.html

Paraschiv, F., Fleten, S.-E., & Schürle, M. (2015). A spot-forward model for electricity prices with regime shifts. *Energy Economics*, *47*, 142–153. https://doi.org/10.1016/j.eneco.2014.11.003

Pozzolo, P. (2023, October 19). *La regressione Lineare Spiegata Semplice*. Paola Pozzolo. https://paolapozzolo.it/regressione-lineare/

Riscaldamento Globale di 1.5 - IPCC - focal point italia. IPCC. (2022, March 30). https://ipccitalia.cmcc.it/ipcc-special-report-global-warming-of-1-5-c/

Team, D. (2023, October 30). *NumPy : The most used Python Library in Data Science*. DataScientest. https://datascientest.com/en/numpy-the-python-library-in-data-science#:~:text=Among%20the%20many%20libraries%20in,a%20variety%20of%20mathematical%20functions

Terna. (2022, June 1). *Le cinque cose da sapere quando si parla di uvam | lightbox*. lightbox. https://lightbox.terna.it/it/insight/progetti-pilota-uvam

Unità virtuali abilitate Miste (UVAM): Vantaggi per i Consumatori: Enel X store Italy. Enel X Store . (n.d.). https://www.enelxstore.com/it/it/news-blog/vantaggi-unita-virtuali-abilitate-miste

Università degli Studi di Trieste. (n.d.-c). Il mercato elettrico italiano (IPEX). https://moodle2.units.it/pluginfile.php/493505/mod_resource/content/0/06_Mercato%20italiano.pdf

Verde, R. (n.d.). *Modello classico di regressione lineare*. Università degli Studi di Napoli Federico II. https://www.docenti.unina.it/webdocenti-be/allegati/materiale-didattico/138303

Wacket, M., Abnett, K., & Alkousaa, R. (2023, March 21). Exclusive: eu drafts plan to allow e-fuel combustion engine cars. (R. More & M. Potter, Eds.)*Reuters*.

Wagner, A., Ramentol, E., Schirra, F., & Michaeli, H. (2022). Short- and long-term forecasting of electricity prices using embedding of calendar information in Neural Networks. *Journal of Commodity Markets*, *28*, 100246. https://doi.org/10.1016/j.jcomm.2022.100246

What is numpy?#. What is NumPy? - NumPy v1.26 Manual. (n.d.). https://numpy.org/doc/stable/user/whatisnumpy.html

What is pandas python?. NVIDIA Data Science Glossary. (n.d.). https://www.nvidia.com/en-us/glossary/data-science/pandas-python/

What is Python. Menu. (n.d.). https://developer.oracle.com/it/learn/technical-articles/what-is-python

Wikimedia Foundation. (2023, November 3). *Python*. Wikipedia. https://it.wikipedia.org/wiki/Python

Appendice

Codice Python utilizzato:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Creazione Dataframe
file_path = 'C:\\Users\\ASUS\\Desktop\\PREZZI ENERGIA\\PUN 2005_2010.xlsx'
df = pd.read_excel(file_path)
data=df.copy()

# Creazione dummies
pd.get_dummies(data, columns=['Mese'])
pd.get_dummies(data, columns=['Giorno_Settimana']) # Queste 2 non servono

# Per le ore bisogna combinare l'ora con il mese ed il giorno della settimana
data['Comb']=data['Ora'].astype(str) + '_' + data['Giorno_Settimana'] + '_' + data['Mese']

dummy_ora=pd.get_dummies(data['Comb'])
data=pd.concat([data,dummy_ora], axis=1)

# Python ci da come risultato delle dummies un True o False quindi decido di cambiare questi valori con 1.0 e 0.0 come float in modo tale da non dare problemi in passaggi successivi

data = data.replace({True: 1.0, False: 0.0})

print(data)



# Analisi dei minimi quadrati
# Per creare la X utile per il metodo dei minimi quadrati dal dataframe uso tutte le colonne tranne quelle specificate
```

```
X = data.drop(['Anno', 'Mese', 'Giorno_Mese', 'Giorno_Settimana', 'Ora', 'PUN', 'Comb'], axis=1).values

# Ad Y metto la colonna PUN
Y = data['PUN'].values

# Faccio la trasposta della matrice X
X_T = X.transpose()

# Faccio il prodotto scalare tra la matrice X e la matrice trasposta X_T
X_X = np.dot(X_T, X)

# Faccio il prodotto scalare tra la matrice Y e la matrice trasposta X_T
X_Y = np.dot(X_T, Y)

# Creazione del modello lineare e dei dati
coefficients = np.linalg.solve(X_X, X_Y)
coeff_data = pd.DataFrame({'Coefficient': coefficients})

# Manipolazione dataframe
# Rimozione di alcune colonne
confronto = data.drop(['Anno', 'Mese', 'Giorno_Mese', 'Giorno_Settimana', 'Ora', 'Comb'], axis=1)

# Inserimento della colonna PUN alla fine del dataset
confronto = confronto[[col for col in confronto.columns if col != 'PUN'] + ['PUN']]

# Creazione di un dataframe con una riga vuota
riga_vuota = {col: None for col in confronto.columns}

# Aggiunta della riga vuota al dataframe confronto
confronto = pd.concat([pd.DataFrame([riga_vuota]), confronto], ignore_index=True)

# Inserimento in questa prima riga vuota dei coefficienti dei minimi quadrati
confronto.iloc[0, :len(coefficients)] = coefficients
```

```
# Inserimento della colonna PUN_stag
confronto['PUN_stag'] = None


# Calcolo della somma pesata per ottenere i valori del PUN_stag
colonne_numeriche = confronto.columns.difference(['PUN', 'PUN_stag'])
confronto['PUN_stag'] = confronto.apply(lambda row: np.sum(row[colonne_numeriche] * confronto.loc[0, colonne_numeriche]), axis=1)

# Creazione di un dataframe con queste colonne
data_anno = data[['Anno', 'Mese', 'Giorno_Mese', 'Giorno_Settimana', 'Ora']].copy()

# Stampa il dataframe data_anno
print(data_anno)

# Creazione di una riga vuota
riga_vuota = pd.DataFrame([{}], columns=data_anno.columns)

#Inserimento della riga vuota nel dataframe data_anno
data_anno = pd.concat([riga_vuota, data_anno], ignore_index=True)

# Stampa il dataframe data_anno
print(data_anno)


# Unisci i dataframe confronto e data_anno
risultato_finale = pd.concat([data_anno, confronto], axis=1)

# Stampa il risultato finale
print(risultato_finale)

# Grafici di confronto PUN

# Filtra i dati nel range specificato
filtered_data = risultato_finale[
```

```
    (risultato_finale['Anno'] == 2008) &
    (risultato_finale['Mese'] == 'Novembre') &
    (risultato_finale['Giorno_Mese'].between(1, 30)) &
    (risultato_finale['Ora'] == 12)
]

# CREA GRAFICI A LINEE

# Creazione di una figura di 10 unità in larghezza e 6 in altezza
plt.figure(figsize=(10, 6))
# Creazione grafici di PUN e PUN_stag rispetto all'intervallo indicato
plt.plot(filtered_data['PUN'], label='PUN', marker='o')
plt.plot(filtered_data['PUN_stag'], label='PUN_stag', marker='o')

# Aggiungi titoli e legende
plt.title('Confronto tra PUN e PUN_stag')
plt.xlabel('Indice')
plt.ylabel('Valore')
plt.legend()

# Mostra il grafico
plt.show()
```