

Guida all'utilizzo del modello di analisi di fattibilità tecno-economica di una Comunità Energetica Rinnovabile sviluppato da ALMACIS

Sommario

1. Introduzione	2
2. File e cartelle	3
3. Installazione di ANACONDA e dei pacchetti necessari alla simulazione	3
Installazione dei pacchetti (“packages”) necessari	5
4. Impostazione di una simulazione	6
Informazioni su Jupyter	6
Creazione dei file input consumo carichi e produzione fotovoltaica – risoluzione oraria	7
Modifica degli input del modello.....	9
5. Risultati	11

1. Introduzione

Questo report costituisce una breve guida all'utilizzo del modello per simulare la fattibilità tecno-economica di una Comunità Energetica Rinnovabile (CER) o di un Gruppo di Auto consumatori che agiscono Collettivamente. Il modello è stato sviluppato dall'azienda ALMACIS utilizzando la piattaforma *Jupyter notebook* unita al linguaggio di programmazione *Python*.

Il modello si basa sull'integrazione della produzione oraria di un sistema fotovoltaico (FV) e dei consumi aggregati di tutte le possibili utenze della CER. C'è la possibilità di integrare anche un sistema di accumulo. Il diagramma di flusso del modello è mostrato in Figura 1.

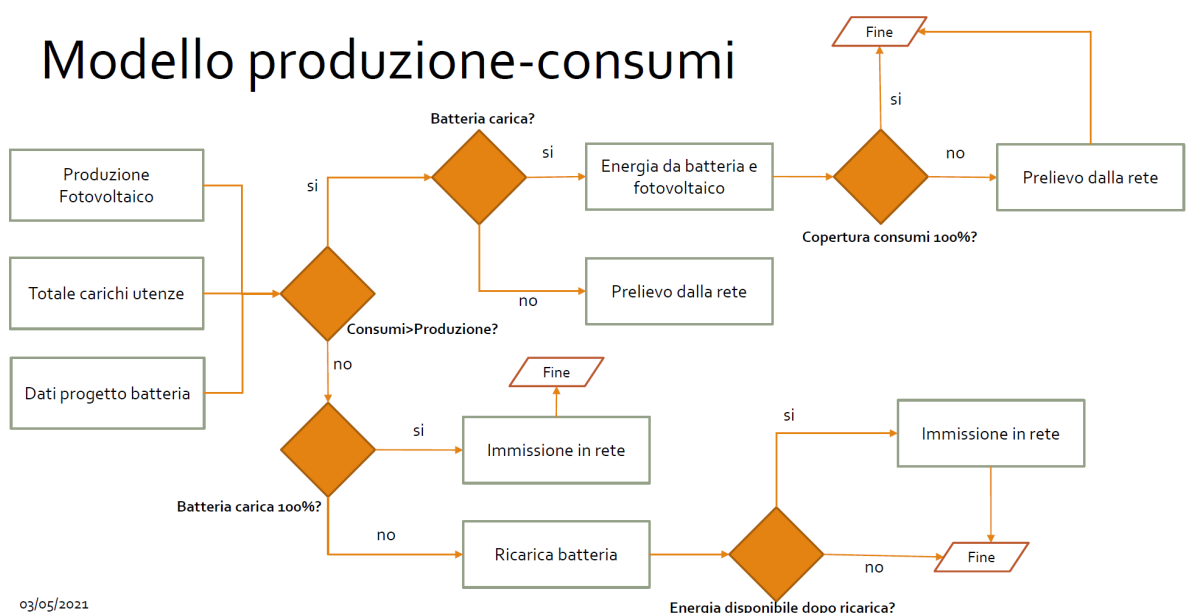


Figura 1: Diagramma di flusso del modello

I dati di INPUT necessari alla simulazione sono:

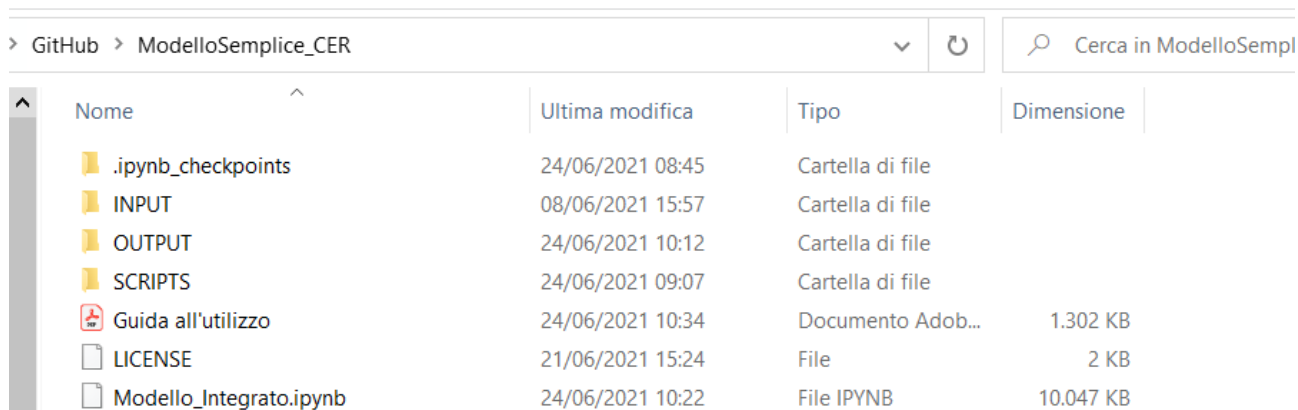
- 1) I dati orari di produzione dell'impianto fotovoltaico che si vuole installare
- 2) I dati orari del totale dei carichi. Questi dati devono quindi includere la somma di tutte le utenze della CER con risoluzione oraria
- 3) I dati di progetto del sistema di accumulo

È importante sottolineare che allo stato attuale di sviluppo del modello non è stato implementato alcun algoritmo per il calcolo dei consumi orari di varie tipologie di utenze a partire da dati aggregati (esempio mensili, o associati a tipiche fasce di consumo F1-F2-F3).

Questo strumento è in grado di effettuare simulazioni solamente se vengono dati come input i dati orari, quindi il vettore con gli 8760 punti annuali dei consumi delle varie utenze.

2. File e cartelle

Il modello è contenuto all'interno della cartella "ModelloSemplice_CER.zip". Per poterlo usare, è necessario estrarre i file e metterli dentro una cartella in una directory locale del proprio computer. In Figura 6 è mostrato un esempio in cui i file sono stati estratti in una cartella "ModelloSemplice_CER" in una sotto-cartella della directory "Documenti".



GitHub > ModelloSemplice_CER				
Nome	Ultima modifica	Tipo	Dimensione	
.ipynb_checkpoints	24/06/2021 08:45	Cartella di file		
INPUT	08/06/2021 15:57	Cartella di file		
OUTPUT	24/06/2021 10:12	Cartella di file		
SCRIPTS	24/06/2021 09:07	Cartella di file		
Guida all'utilizzo	24/06/2021 10:34	Documento Adob...	1.302 KB	
LICENSE	21/06/2021 15:24	File	2 KB	
Modello_Integrato.ipynb	24/06/2021 10:22	File IPYNB	10.047 KB	

Figura 2: File necessari per far girare il modello

Per dare una panoramica della struttura:

- 1) Il file "Modello_Integrato.ipynb" è il file notebook principale tramite cui il modello viene fatto girare. Questo file andrà successivamente aperto tramite Jupyter – verrà spiegato successivamente nel report come eseguire questa operazione
- 2) La cartella "SCRIPTS" contiene le funzioni necessarie per far girare il modello. A meno che non si voglia entrare nel merito del codice di simulazione, questa cartella non va modificata
- 3) "INPUT" è la cartella che deve contenere i file "csv" dove vengono inseriti gli input necessari per il modello, quindi i dati orari dei consumi di tutti i membri della CER e i dati orari di producibilità dell'impianto fotovoltaico.
- 4) "OUTPUT" è la cartella dove vengono salvati i file contenenti alcuni risultati principali delle varie simulazioni
- 5) Il file "Guida all'utilizzo" contiene una versione PDF di questo report, mentre LICENSE specifica le condizioni di licenza e responsabilità.

3. Installazione di ANACONDA e dei pacchetti necessari alla simulazione

Per potere effettuare una simulazione usando Jupyter sul proprio computer, è necessario installare Python 3.8 e una piattaforma di distribuzione tramite cui lanciare il programma. Una delle opzioni è l'installazione di Anaconda, una piattaforma open-source di distribuzione di Python che semplifica tanti passaggi, fra cui l'installazione di pacchetti e software che sono in grado di leggere Python.

È possibile effettuare il download di Anaconda al seguente indirizzo web:

<https://www.anaconda.com/products/individual>

Io consiglio l'installazione in locale, ovvero del solo utente del PC che intende utilizzare Anaconda.

Una volta installato Anaconda, è possibile accedere al "Navigator" (cercare Anaconda Navigator) tramite cui è possibile aprire alcuni dei software installati con il pacchetto. L'interfaccia di Anaconda Navigator è

mostrata in Figura 2. Come si vede dalla figura, è possibile lanciare l'applicazione Jupyter dal Navigator cliccando sul pulsante "Launch".

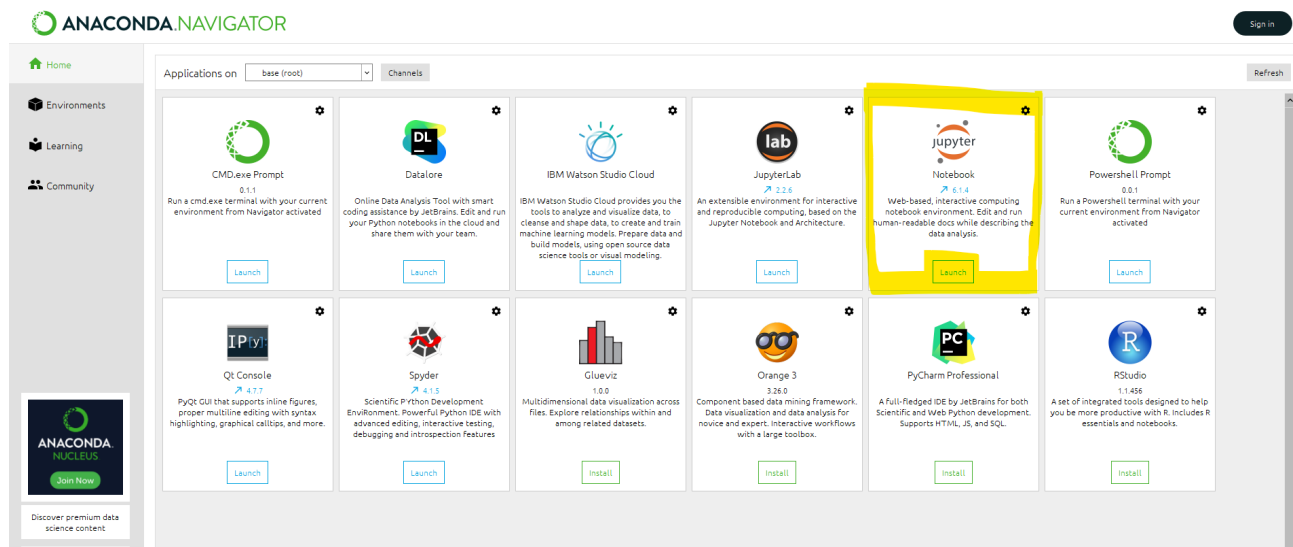


Figura 3: Interfaccia di ANACONDA navigator

Una volta cliccato su "Launch", si verrà indirizzati alla cartella "locale" (ad esempio io sono indirizzata C:\Users\roberta.mancini) sempre tramite Jupyter notebook. (come si vede dalla Figura 3).

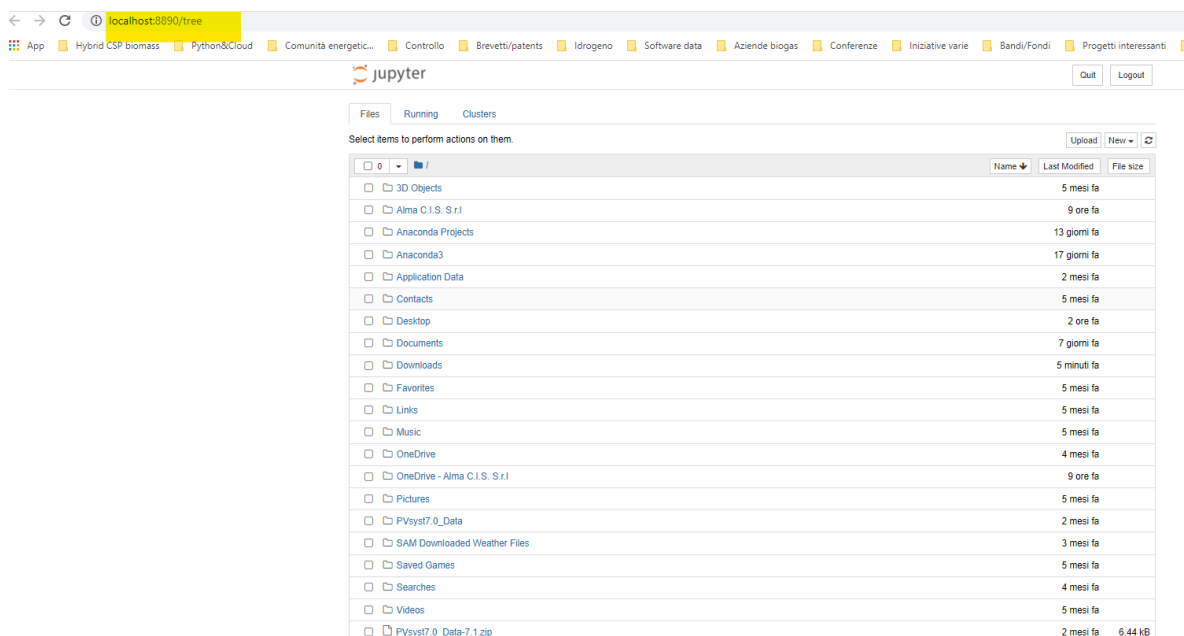


Figura 4: Interfaccia Jupyter – 1

Navigando nelle varie cartelle locali è quindi possibile creare o aprire file esistenti tramite Jupyter. L'estensione dei cosiddetti notebook è data da `.ipynb`, mentre l'estensione di file Python tradizionali è data da `.py`.

È quindi possibile copiare la cartella contenente il modello in una delle directory locali, e aprire il file principale dall'interfaccia.

Installazione dei pacchetti (“packages”) necessari

Il modello è basato sull'utilizzo di alcuni pacchetti (chiamati nel linguaggio informatico “packages”) che vanno preliminarmente installati tramite Anaconda.

Per fare ciò, occorre lanciare il cosiddetto “Anaconda prompt” (il terminal viene automaticamente installato quando viene scaricato Anaconda). Si lancia il terminal come mostrato in Figura 4, e si possono inserire i comandi necessari per l'installazione dei packages.

Nel prompt mostrato in Figura 4 (sulla destra) vanno scritti i seguenti comandi per l'installazione di altrettanti pacchetti. Va inserito un comando alla volta, e una volta premuto invio si passa all'installazione.

I comandi da inserire e i pacchetti da installare sono i seguenti:

- conda install numpy
- conda install pandas
- conda install plotly

Un esempio è riportato in Figura 5.

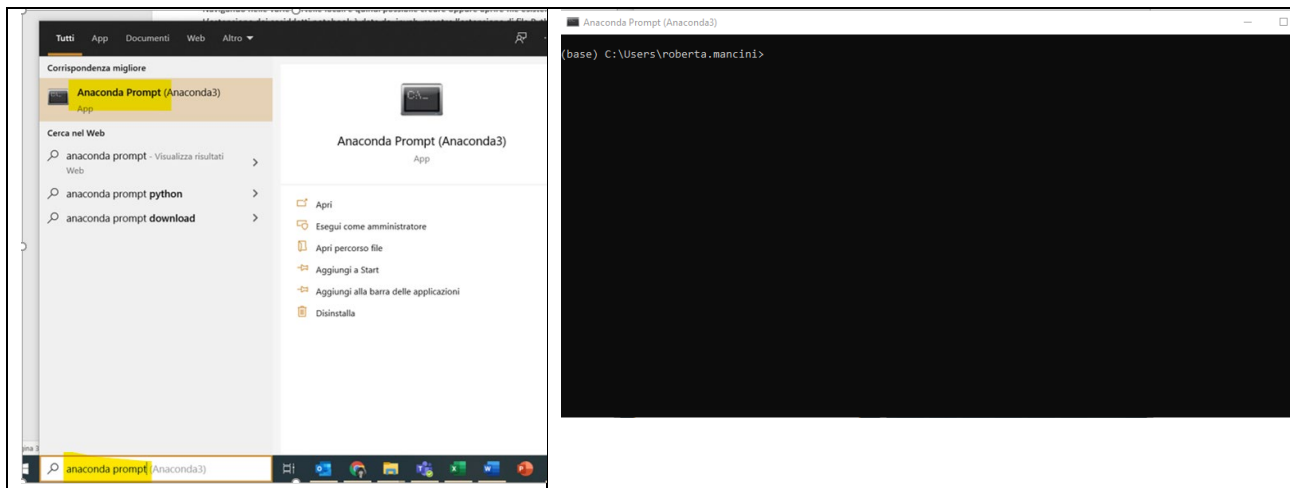


Figura 5: Anaconda prompt (a sx: lancio del prompt tramite la barra windows, a dx: vista del prompt all'apertura)

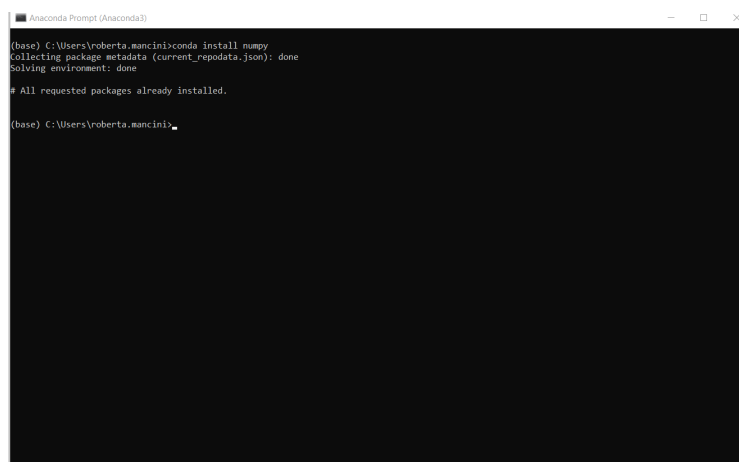


Figura 6: esempio di installazione di pacchetto tramite Anaconda prompt

4. Impostazione di una simulazione

Informazioni su Jupyter

Una volta installato correttamente Anaconda e scaricati i file contenuti nella cartella del modello in una cartella locale (qualsiasi), è possibile aprire file principale “Modello_Integrato.ipynb” tramite Jupyter.

- Lanciare Jupyter notebook come mostrato in Figura 3.
- Trovare il percorso corrispondente alla cartella modello, e aprire il file “Modello_Integrato.ipynb” come mostrato in Figura 7

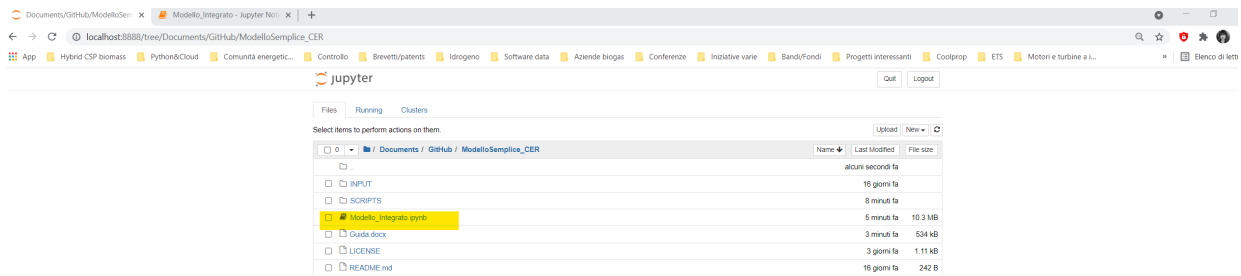


Figura 7: Interfaccia Jupyter – 2

Il file dovrebbe avere un aspetto come mostrato in Figura 8.

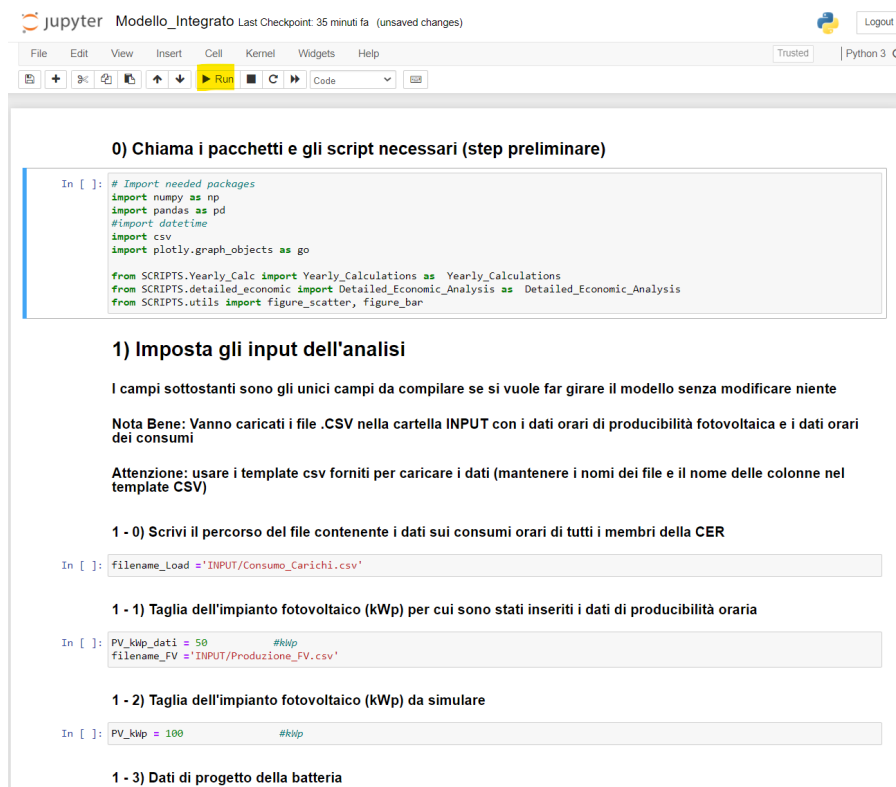


Figura 8: File principale del modello in Jupyter - 1

Ogni box (cella) grigio corrisponde a delle linee di “codice” che è possibile far girare. Iniziando dal primo box si può cliccare sul pulsante “Run” evidenziato in giallo in Figura 8 per far girare il codice. Per lanciare una simulazione completa bisogna far girare tutte le celle del notebook partendo dalla prima e scendendo via

via. Una volta che una cella è stata fatta girare, un numero apparirà sulla sinistra (come mostrato in Figura 9) e corrisponderà all'ordine in cui le varie celle sono state lanciate.

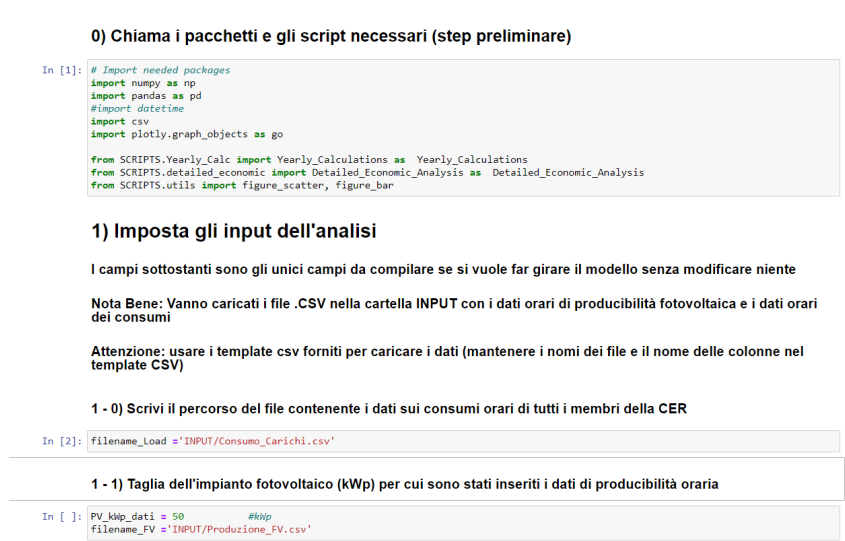


Figura 9: File principale del modello in Jupyter - 2

Invece di far girare le celle una per volta, è possibile lanciarle tutte insieme premendo il pulsante con la doppia freccia mostrato in Figura 10, e successivamente scegliendo l'opzione "Restart and Run All Cells".

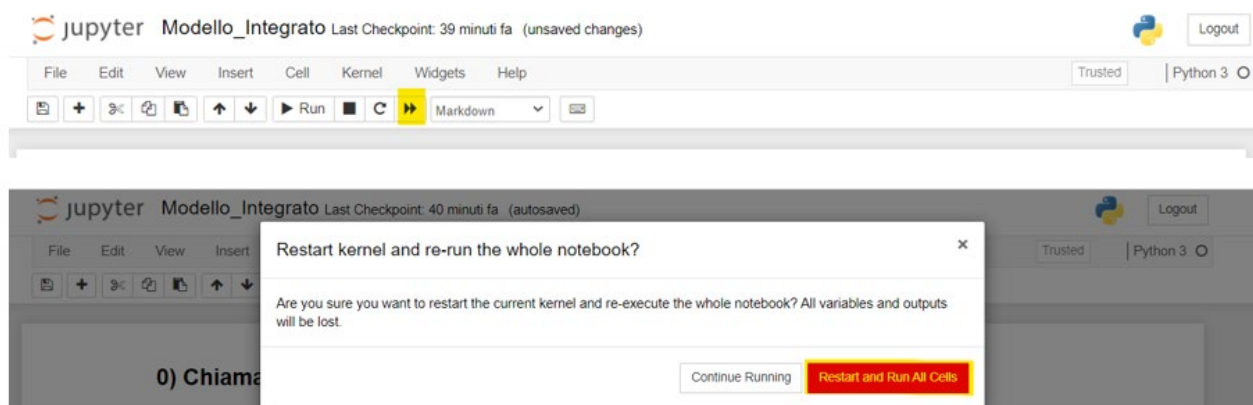


Figura 10: Istruzioni su come far girare tutte le celle

In questo modo, dopo aver impostato correttamente gli input, è possibile far girare l'intero modello con un solo click.

Creazione dei file input consumo carichi e produzione fotovoltaica – risoluzione oraria

Come anticipato preliminarmente è necessario caricare dei file di input contenenti i dati orari dei carichi e della produzione di un pannello fotovoltaico. Qui è assolutamente importante usare i due file che sono già presenti nella cartella come TEMPLATE per inserire i propri dati. Il file con i consumi deve contenere

necessariamente una colonna chiamata “Consumo carichi, kWh” al di sotto della quale vanno inseriti i consumi orari in kWh. La colonna deve contenere 8760 punti.

Il file con i dati di producibilità invece deve contenere i dati di producibilità di un dato impianto FV sotto una colonna chiamata “Produzione FV, kWh”, che deve contenere anch’essa 8760 punti. **Attenzione. Non serve inserire i dati di producibilità della taglia di FV che si vuole necessariamente simulare. Basta inserire i dati per una certa taglia in kWp, e poi vedremo come è possibile scalare la produzione nel modello.**

Si consiglia di usare i due file già forniti senza modificare i nomi delle colonne in modo da non commettere errori. Figura 11 mostra i due file. **Attenzione! I dati vanno inseriti con Excel impostato ad avere “,” come separatore decimale.**

The figure displays two Excel spreadsheets side-by-side, both with the 'Home' ribbon selected. The left spreadsheet is titled 'Produzione_FV' and the right is 'Consumo_Carichi'.

Produzione_FV (Left): The spreadsheet has columns A through I. Row 1 has 'Ora' in A and 'Produzione FV, kWh' in B. Rows 2-10 show data for hours 1-9, with production values of 0 for hours 1-8 and 0,8655 for hour 9. Row 10 shows a total production of 6,754590909.

Ora	Produzione FV, kWh
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0,8655
10	6,754590909

Consumo_Carichi (Right): The spreadsheet has columns A through I. Row 1 has 'Ora' in A and 'Consumo carichi, kWh' in B. Rows 2-12 show data for hours 1-11, with consumption values ranging from 3,25 to 48,375 kWh.

Ora	Consumo carichi, kWh
1	48,375
2	3,75
3	6,25
4	3,25
5	6,5
6	3,25
7	11
8	38,75
9	14,125
10	3,875
11	11,25

Figura 11: File CSV contenenti i dati orari – template

Modifica degli input del modello

Per modificare tutti gli input del modello, è necessario modificare SOLAMENTE i valori contenuti nelle celle mostrate in Figura 12, che corrispondono alla sezione **1) Imposta gli input dell'analisi del notebook**.

1) Imposta gli input dell'analisi

I campi sottostanti sono gli unici campi da compilare se si vuole far girare il modello senza modificare niente

Nota Bene: Vanno caricati i file .CSV nella cartella INPUT con i dati orari di producibilità fotovoltaica e i dati orari dei consumi

Attenzione: usare i template csv forniti per caricare i dati (mantenere i nomi dei file e il nome delle colonne nel template CSV) ¶

1 - 0) Scrivi il percorso del file contenente i dati sui consumi orari di tutti i membri della CER

```
In [2]: filename_Load = 'INPUT/Consumo_Carichi.csv'
```

1 - 1) Taglia dell'impianto fotovoltaico (kWp) per cui sono stati inseriti i dati di producibilità oraria

```
In [3]: PV_kWp_dati = 50          #kWp
filename_FV = 'INPUT/Produzione_FV.csv'
```

1 - 2) Taglia dell'impianto fotovoltaico (kWp) da simulare

```
In [4]: PV_kWp = 100            #kWp
```

1 - 3) Dati di progetto della batteria

```
In [5]: Capacity_Battery = 0      # Capacità, kWh
Maximum_Power = 0                # Potenza massima erogabile dalla batteria, kW
SOC_min = 5                      # Stato di carica minimo, %
eta_BC = 0.99                    # Efficienza di carica
eta_BD = 0.99                    # Efficienza di scarica
SOC_0 = 50                       # Stato di carica iniziale, %
```

1 - 4) Input economici

```
In [6]: # Dati investimento - COSTO DI INVESTIMENTO SPECIFICO
CAPEX_PV_specifico = 1200        # eur/kWp
CAPEX_battery_specifico = 500    # eur/kWh

# Dati costi operativi (espressi in % dell'investimento)
OPEX_Perc = 3.5                  # %
Inverter = 10                    # %

# Scegli la tipologia di schema
# Opzioni 'AC', 'CER'
Tipo = 'CER'                     #'CER' oppure 'AC'

# Dati incentivo (Lasciare invariato se si vuole rispettare la normativa attuale)
Incentivo_MISE = 0.11 * (Tipo=='CER') + 0.1 * (Tipo=='AC') # eur/kWh
Sgravio_ARERA = 0.008 * (Tipo=='CER') + 0.01 * (Tipo=='AC') # eur/kWh

# Prezzo di mercato dell'energia elettrica
Prezzo_El_Mercato = 0.05         # eur/kWh

# Tasso di attualizzazione - necessario per il calcolo del Valore Attuale Netto
Tasso_Attualizzazione = 5        # %
```

Figura 12: Celle del notebook che vanno modificate per impostare gli input

Seguendo l'ordine che compare su Jupyter è possibile modificare:

- 1) Il percorso/nome del file contenente i dati dei consumi carichi **"filename_Lod"**, che ora è impostato su 'INPUT/Consumo_Carichi.csv'. Se si salva il file nella cartella INPUT con un altro nome, è necessario inserire il nome giusto.
- 2) Il percorso/nome del file contenente i dati di producibilità oraria **"filename_FV"**, che ora è impostato su 'INPUT/Produzione_FV.csv'. Se si salva il file nella cartella INPUT con un altro nome, è necessario inserire il nome giusto.
- 3) **PV_kWp_dati** che rappresenta la taglia di impianto fotovoltaico per cui sono stati inseriti i dati (cioè la taglia a cui fa riferimento il file .csv).
- 4) **PV_kWp** è la taglia dell'impianto che invece si vuole simulare (è diversa dalla taglia per cui vengono inseriti i dati, la prodicibilità si scala proporzionalmente considerando un campo con le stesse caratteristiche)
- 5) I dati di progetto della batteria, mostrati qui:

1 - 3) Dati di progetto della batteria

```
Capacity_Battery = 0          # Capacità, kWh
Maximum_Power = 0             # Potenza massima erogabile dalla batteria, kW
SOC_min = 5                   # Stato di carica minimo, %
eta_BC = 0.99                 # Efficienza di carica
eta_BD = 0.99                 # Efficienza di scarica
SOC_0 = 50                    # Stato di carica iniziale, %
```

- 6) Gli input economici, mostrati di seguito.

1 - 4) Input economici

```
In [6]: # Dati investimento - COSTO DI INVESTIMENTO SPECIFICO
CAPEX_PV_specifico = 1200      # eur/kWh
CAPEX_battery_specifico = 500  # eur/kWh

# Dati costi operativi (espressi in % dell'investimento)
OPEX_Perc = 3.5                # %
Inverter = 10                  # %

# Scegli la tipologia di schema
# Opzioni 'AC', 'CER'
Tipo = 'CER'                   #'CER' oppure 'AC'

# Dati incentivo (Lasciare invariato se si vuole rispettare la normativa attuale)
Incentivo_MISE = 0.11 * (Tipo=='CER') + 0.1 * (Tipo=='AC') # eur/kWh
Sgravio_ARERA = 0.008 * (Tipo=='CER') + 0.01 * (Tipo=='AC') # eur/kWh

# Prezzo di mercato dell'energia elettrica
Prezzo_El_Mercato = 0.05       # eur/kWh

# Tasso di attualizzazione - necessario per il calcolo del Valore Attuale Netto
Tasso_Attualizzazione = 5      # %
```

Vanno inserirti i dati di investimento specifico FV e accumulo, la % dei costi operativi, la % del costo dell'inverter, il tipo di schema "CER" o "AC" (autoconsumo collettivo), i valori degli incentivi, il prezzo di mercato dell'energia elettrica, e il tasso di attualizzazione necessario per il calcolo del Valore Attuale Netto (VAN).

5. Risultati

A livello di numeri e grafici molti risultati sono direttamente visualizzabili sul notebook. Un esempio è riportato in Figura 13. Attenzione! Si ricorda che è possibile zoomare su tutti i grafici se si vuole aumentare la risoluzione dei grafici orari per un intero anno per visualizzare, ad esempio, un paio di giorni. Basta cliccare con il cursore sull'area da ingrandire.

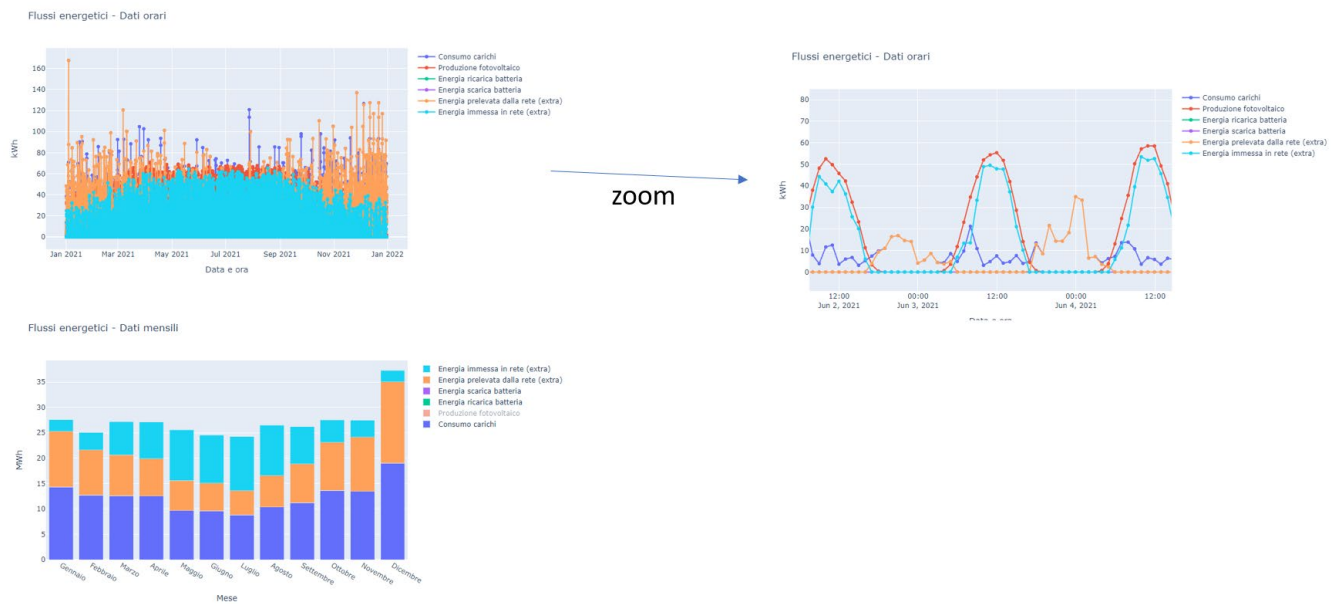


Figura 13: Alcuni risultati del modello

Alla fine della simulazione, alcuni risultati principali vengono esportati in un file csv, contenuto nella cartella OUTPUT. Se si vuole modificare il nome del file (per esempio per non sovrascrivere file di altre simulazioni precedentemente salvati) si può modificare il campo “filename_output” nelle ultime celle del notebook.

5) Esporta i risultati in un file CSV

```
In [19]: # Scegli il nome del file in cui salvare i risultati
filename_output = 'OUTPUT/OutputSimulazione.csv'
```

```
In [20]: OUTPUT_DICTIONARY = {'Taglia FV kWp': PV_kWp,
                              'Taglia batteria kWh': Capacity_Battery,
                              'Consumi annuali kWh/anno': economy_output['Energia_Consumi'],
                              'Investimento FV EUR': economy_output['CAPEX_PV'],
                              'Investimento accumulo EUR': economy_output['CAPEX_batteria'],
                              'Costi operativi EUR/anno': economy_output['OPEX'],
                              'Costi riampio inverter EUR/anno': economy_output['InverterCost'],
                              'Energia condivisa FV kWh/anno': economy_output['Energia_Condivisa_PV'],
                              'Energia condivisa batteria kWh/anno': economy_output['Energia_Condivisa_Batteria'],
                              'Totale energia immessa in rete kWh/anno': economy_output['Energia_Immessa_In_Rete'],
                              'Totale energia prelevata dalla rete kWh/anno': economy_output['Energia_Prelevata_tot'],
                              'Ricavo MISE EUR/anno': economy_output['Ritorno_MISE'],
                              'Sgravio ARERA EUR/anno': economy_output['Ritorno_ARERA'],
                              'Vendita energia elettrica EUR/anno': economy_output['Ritorno_RID'],
                              'Tempo di ritorno senza detrazioni': PBT_1,
                              'Tempo di ritorno con detrazioni': PBT_detrazioni,
                              }

df_OUTPUT = pd.DataFrame.from_dict(OUTPUT_DICTIONARY,orient='index')
df_OUTPUT.to_csv(filename_output, decimal=',', sep=';')
```

```
In [ ]:
```