



UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA
Corso di Laurea Triennale in Ingegneria Elettronica

Identificazione di prosumer solari utilizzando i dati degli smart meter

Solar prosumer identification using net smart meter data

Relatore:

Prof. Emanuele Principi

Tesi di laurea di:

Kurosh Ameri

Anno accademico 2019/2020

Abstract

La distribuzione decentralizzata dell'energia elettrica ha portato negli anni ad un aumento del numero di utenti in grado sia di consumare che di produrre energia tramite generazione individuale da impianti fotovoltaici. Sfortunatamente, i gestori della rete non sono a conoscenza dell'ubicazione di tutti i produttori di energia solare a causa di installazioni non autorizzate o non dichiarate.

Conoscere l'ubicazione dei prosumers solari può informare la protezione del circuito e le impostazioni di regolazione della tensione, e aiutare i gestori di rete a pianificare meglio le variazioni giornaliere della domanda energetica, che vengono ulteriormente amplificate dall'intermittenza della generazione.

Il presente lavoro di tesi si pone l'obiettivo di definire ed implementare un algoritmo machine learning non supervisionato capace di identificare i clienti generatori di energia fotovoltaica utilizzando i dati di consumo energetico netto forniti degli smart meter.

Metodi di riduzione della dimensionalità sono valutati al fine di proporre un mezzo efficace per ridurre la quantità di dati necessari all'identificazione e migliorare il tempo di clustering di un fattore significativo.

L'approccio di classificazione avanzata proposto per l'identificazione dei prosumer solari, basato sul clustering agglomerativo, dimostra un'accuratezza di classificazione molto significativa e supera le prestazioni del clustering K-Mean.

Indice

Capitolo 1, Introduzione:	1
1.1 Prolifcazione dei prosumer solari	1
1.2 Necessità di identificare i prosumer	3
Capitolo 2, Descrizione dell' algoritmo	6
2.1 Data cleaning	7
2.2 Data transformation	8
2.3 Dimensionality reduction	9
2.4 Metodi di clustering	12
2.5 Etichettatura dei cluster	15
Capitolo 3, Analisi	16
3.1 Descrizione dei dati	16
3.2 Data cleaning	18
3.3 Risultati del clustering	19
3.4 Selezione del modello di classificazione	21
Capitolo 4, Conclusioni e possibili applicazioni dell' algoritmo	27
Appendice A	29
Appendice B	30
Appendice C	31
Riferimenti	33
Elenco delle figure	35
Ringraziamenti	37

CAPITOLO 1

Introduzione

1.1 Proliferazione dei prosumer solari

Nel settore energetico si indica con il termine “prosumer” il cliente capace di agire sia come produttore che come consumatore di energia elettrica. Questo segmento di utenti della rete è cresciuto notevolmente, di pari passo con l'aumento della generazione decentralizzata di energia elettrica.

La recente proliferazione del fotovoltaico ha reso l'autoproduzione una prospettiva fattibile per molti consumatori residenziali. I governi di tutto il mondo forniscono incentivi per installazioni domestiche di fotovoltaico: in Italia, ad esempio, già da diversi anni chi acquista un impianto solare per casa può usufruire di una detrazione fiscale del 50%, valida sia per case singole che per condomini. [1]

Secondo i dati contenuti nel Rapporto statistico sul Solare fotovoltaico 2019 [2], pubblicato lo scorso giugno dal Gse, attualmente nel nostro paese risultano installati oltre 880.090 impianti fotovoltaici, per una potenza totale di circa 21 GW e una produzione attorno ai 24 TWh. Durante l'anno scorso sono stati realizzati più di ottantacinquemila nuovi impianti fotovoltaici, i quali hanno incrementato di 750 MW la potenza installata del paese: un trend di crescita quasi doppio rispetto ai 400 MW degli anni precedenti.

Le nuove installazioni hanno visto la nascita di impianti con potenza inferiore ai 20 kW – segnale che indica una crescente diffusione dell'autoconsumo – ma anche una

ripresa degli investimenti sugli impianti di grande taglia, oltre i 5 MW (quasi il 30% della nuova potenza fotovoltaica).

Su un totale di 115 TWh di energia elettrica prodotta da fonti rinnovabili in Italia nel 2019, il fotovoltaico copre una quota pari al 20%, registrando un incremento del 4,6% rispetto all'anno precedente. Tutto questo significa che nel settore elettrico italiano, ogni 10kWh prodotti da fonti rinnovabili, 2kWh di questi provengono dal fotovoltaico.

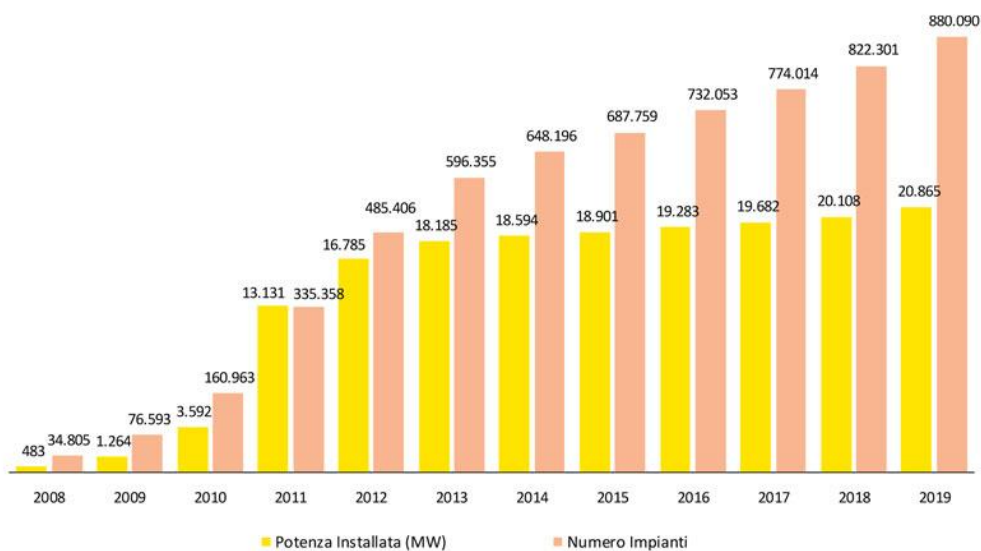


Figura 1.1.1: Gli impianti installati in Italia tra il 2008 ed il 2019

Sul tema dell'autoconsumo fotovoltaico, in Italia gli autoconsumi del 2019 risultano pari quasi al 20% della produzione complessiva degli impianti fotovoltaici e al 39% della produzione degli impianti che solamente autoconsumano.

Il massimo livello di autoconsumo è registrato nel mese di luglio, mentre le quote di autoconsumo più elevate si rilevano in corrispondenza dei mesi invernali.

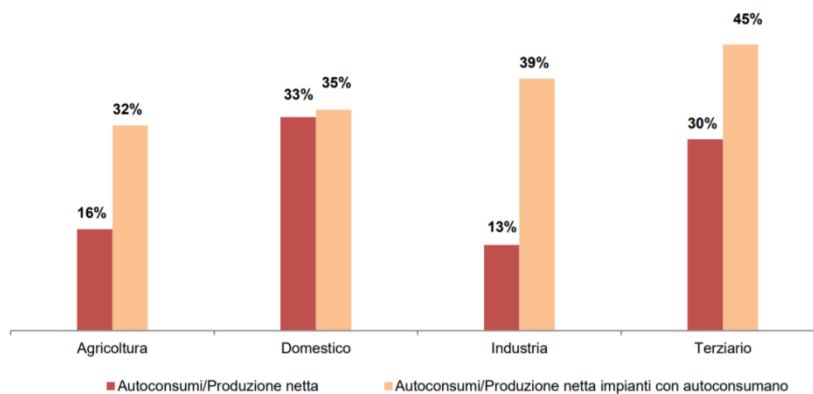


Figura 1.1.2: Profili di autoconsumo per settore di attività nel 2019

Come mostrato in Figura 1.1.2, la quasi totalità degli impianti nel settore domestico in Italia è in autoconsumo. I risultati migliori arrivano dai settori industriale e terziario, che hanno raggiunto percentuali rispettivamente del 39% e del 45% di energia autoconsumata, sul totale di quella prodotta.

Nel nostro paese si è registrato un notevole incremento dei sistemi di accumulo dell'energia prodotta da impianti fotovoltaici. Alla fine del 2019, si contano quasi 26.000 sistemi, con una potenza di 118 MW. Ciò significa che rispetto al 2018, i sistemi di accumulo hanno registrato nel 2019 una crescita di circa il 50% sia in termini di numerosità, che in termini di potenza.

Nonostante che il tema dell'autoconsumo fotovoltaico sia una delle leve principali per il raggiungimento degli obiettivi di sostenibilità energetica fissati dall'Unione Europea nel Quadro 2030 per il clima [3], le installazioni di risorse energetiche decentralizzate in Italia, come nel resto del mondo, sono collegate tramite schemi di Scambio Sul Posto (SSP) dove la generazione e il carico non sono tipicamente misurati separatamente, rendendo più difficile per gli operatori monitorare la loro presenza e le loro prestazioni.

1.2 Necessità di individuare i prosumer solari

Contrariamente al sistema di trasmissione, dove gli impianti di generazione sono molto più grandi e tipicamente misurati separatamente, gli operatori di distribuzione non

hanno visibilità su dove si trovino tutti gli impianti di generazione distribuita a causa di installazioni non autorizzate o non dichiarate e della mancanza di una misurazione separata.

Alcune ragioni per cui queste installazioni potrebbero non risultare dichiarate sono l'elusione fiscale, i cambiamenti nella regolamentazione, sistemi multipli in un'unica residenza, la mancanza di consapevolezza del proprietario e l'inserimento di dati non corretti [4]. Anche per gli impianti di cui si conosce l'ubicazione, il guasto dell'impianto fotovoltaico può verificarsi senza che il proprietario ne sia a conoscenza.

L'aumento dei livelli di penetrazione del fotovoltaico presso la sede del cliente può modificare sostanzialmente il profilo di carico dell'impianto [5], che è un input fondamentale per le operazioni e la pianificazione a breve termine e può anche modificare il profilo di tensione. Una comprensione accurata del luogo di generazione del fotovoltaico è importante per gli operatori a livello di distribuzione quando si esegue la commutazione automatica, la riconfigurazione del circuito o il ripristino del carico, poiché il vero carico del circuito può essere nascosto agli operatori dalla generazione. Inoltre, la consapevolezza della presenza di risorse energetiche distribuite è fondamentale per un'adeguata protezione e regolazione della tensione degli alimentatori di distribuzione. Pertanto, la comprensione della posizione dei prosumers solari è essenziale per le operazioni e la pianificazione.

Poiché lo scopo di questo documento è quello di dividere i clienti non etichettati in due gruppi (prosumer solari e consumatori), le tecniche di machine learning non supervisionate sono adatte ad affrontare questo problema. Sono considerati due tipi di tecniche di apprendimento non supervisionato: il clustering K-Means ed il clustering gerarchico agglomerativo.

I metodi di clustering possono fornire un mezzo per raggruppare clienti simili per comprendere meglio i loro modelli di consumo e portare il vantaggio di poter lavorare con dati non etichettati. Sebbene il clustering dei profili di consumo di energia elettrica dei clienti residenziali è stata un'area attiva per diversi decenni, con la recente ascesa dei prosumer solari, poca attenzione è stata dedicata allo sviluppo di approcci di clustering per l'identificazione di questi ultimi. Pertanto, questo lavoro propone un approccio innovativo, automatizzato e basato sul clustering per identificare i clienti con sistemi di generazione di energia fotovoltaica utilizzando esclusivamente i loro dati sul consumo

netto di energia. Inoltre vengono proposti approcci di riduzione della dimensionalità, sicché si possa ridurre la quantità di dati necessari per identificare accuratamente i prosumers solari in un unico punto di dati, riducendo i tempi di immagazzinamento e di clustering di un fattore significativo.

Il resto del testo è organizzato come segue: il Capitolo 2 illustra in dettaglio la metodologia di classificazione e i metodi di riduzione della dimensionalità. Il Capitolo 4 fornisce l'analisi dei vari modelli di classificazione per l'identificazione dei prosumer solari utilizzando i dati di consumo energetico effettivo di 73 clienti residenziali del Pecan Street Dataport [7] e dimostra l'accuratezza dell'approccio proposto. Infine, il Capitolo 4 presenta la conclusione e discute ulteriori possibili applicazioni dell'algoritmo.

CAPITOLO 2

Descrizione dell'algoritmo

La Figura 2.1 riassume l'approccio complessivo dell'algoritmo.

Al fine di selezionare il modello migliore di classificazione da utilizzare nell'identificazione dei prosumer solari, ogni fase del processo di modellazione è stata analizzata ed i cluster risultanti sono stati convalidati. Ciò permette di comprendere meglio il valore della normalizzazione e della riduzione della dimensionalità.

In questo capitolo verranno argomentati per ordine tutti i passaggi dell'approccio, partendo dalla pulizia dei dati (*Data Cleansing*) fino alla fase di validazione del clustering che ci permette di selezionare il metodo di classificazione migliore

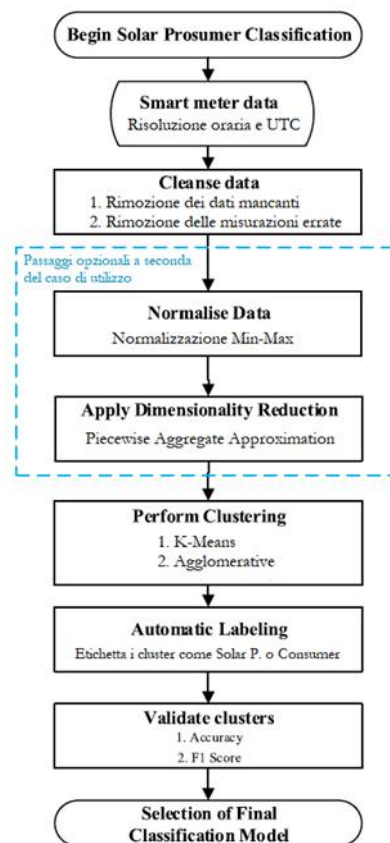


Figura 2.1: Approccio di classificazione per l'identificazione dei prosumer solari

2.1 Data cleansing

La pulizia dei dati è un passaggio necessario per l'implementazione di ogni modello di classificazione. Quando si lavora con dati di serie temporali (*time series data*), i problemi comuni che si presentano sono: dati mancanti, errori nella conversione dei fusi orari, valori ripetuti ed errori del sensore.

Il metodo di data cleansing che è stato utilizzato prende in ingresso dati grezzi dell'energia netta riportati dagli Smart Meter, dopodiché segue la procedura del diagramma riportato in Figura 2.1.2. I passi principali sono la standardizzazione del formato dei dati e la ricerca di dati mancanti, erronei o a linee piatte e la rimozione dei clienti che presentano queste caratteristiche nel periodo di interesse dai dati utilizzati per l'identificazione.

La conversione dei dati in Tempo Coordinato Universale (UTC) elimina l'impatto dell'ora legale consentendo una contabilità coerente tra i diversi fusi orari.

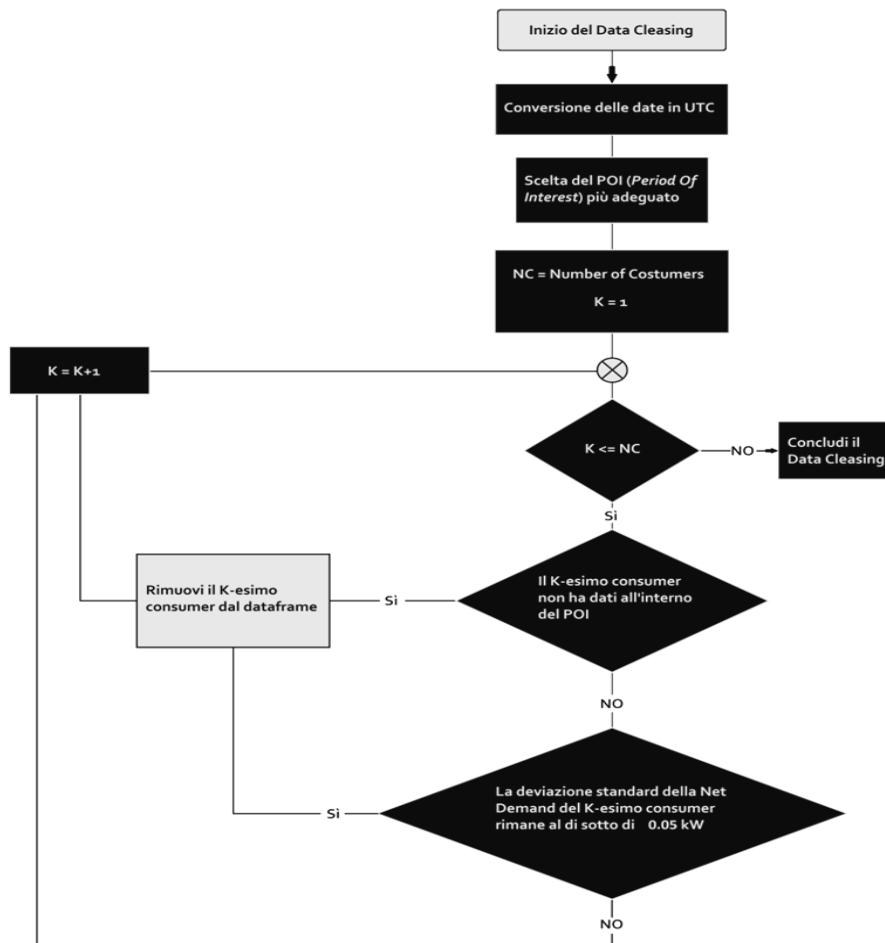


Figura 2.1.2: Processo di data clearing

2.2 Data transformation

La normalizzazione dei dati è una parte essenziale del machine learning quando si lavora con un set di dati di grandi dimensioni. Nel nostro caso, poiché il carico di picco di ogni singolo cliente differisce dall'altro, per confrontare in modo più efficace il carico di ogni cliente e di annullare l'impatto dei diversi offset e delle diverse ampiezze, è importante adoperare una normalizzazione dei dati. L'approccio standard per trasformare i dati è la Normalizzazione Min-Max.

La normalizzazione Min-Max è uno dei modi più comuni per normalizzare i dati. Per ogni caratteristica, il valore minimo di quella caratteristica si trasforma in uno 0, il

valore massimo si trasforma in un 1, e ogni altro valore si trasforma in un decimale tra 0 e 1. La conversione di $X = X_1, X_2, \dots, X_n$ in un intervallo di $[0,1]$ avviene attraverso la seguente formula:

$$x'_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

dove x'_i e x_i sono il consumo di elettricità normalizzato ed effettivo registrato al momento i , e x_{min} e x_{max} sono la domanda minima e massima nel periodo n .

Questa tecnica è altamente interpretabile in quanto mantiene la forma originale del carico.

2.3 Dimensionality reduction

Per l'esecuzione di clustering su una *time series*, un'accurata selezione delle caratteristiche su cui effettuare il clustering può influire sulle prestazioni.

I metodi per ridurre il numero di singoli punti di dati su cui fare il clustering possono avere notevoli vantaggi, fra questi la riduzione dei tempi di elaborazione e la riduzione dei requisiti di archiviazione dei dati.

Per fare questo esistono diversi metodi che vanno dai metodi statistici di aggregazione alla conversione delle serie temporali in uno spettro di frequenza. Nel nostro caso considereremo la *Piecewise Aggregate Approximation* (PAA).

La PAA, come viene ampiamente illustrato in [9], permette la rappresentazione di una *time series* X di lunghezza n in uno spazio dimensionale w da un vettore $\mathbf{X} = x_1, \dots, x_w$ con la dimensionalità ridotta di un fattore n/w , dove x_i è determinato come nella formula seguente:

$$\bar{x}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} x_j$$

La PAA divide le serie temporali in frame equivalenti, calcola la media dei dati all'interno di ogni frame e un vettore della media corrispondente diventa la nuova serie temporale ridotta. Questo metodo di aggregazione porta direttamente ad un'altra rappresentazione chiamata *Symbolic Aggregate approXimation*, abbreviata in SAX, che prende la dimensionalità ridotta e assegna al grafico una rappresentazione di stringa. Fornendo una rappresentazione di stringa, i dati complessivi che devono essere memorizzati sono meno di altri metodi di data mining come la trasformata discreta di Fourier (DFT) [10].

Nelle figure seguenti è riportata a scopo di esempio l'applicazione di PAA sui dati di consumo energetico netto di uno dei residenti documentati nel dataset di Pecan Street. Nell'esempio sono stati utilizzati i dati per un periodo di tempo di una settimana (178 ore), in cui è stata scelta una finestra di $n/w = 6$ per la PAA. In entrambe le figure i risultati della PAA vengono confrontati con i dati non elaborati del consumo energetico, in kilowatt (kW) nella Figura 2.3.1, in valori normalizzati (%) nella Figura 2.3.2.

È interessante osservare che il grafico rimanga pressoché identico nelle due immagini, a conferma di quanto si è detto a riguardo della Normalizzazione Min-Max nel paragrafo precedente.

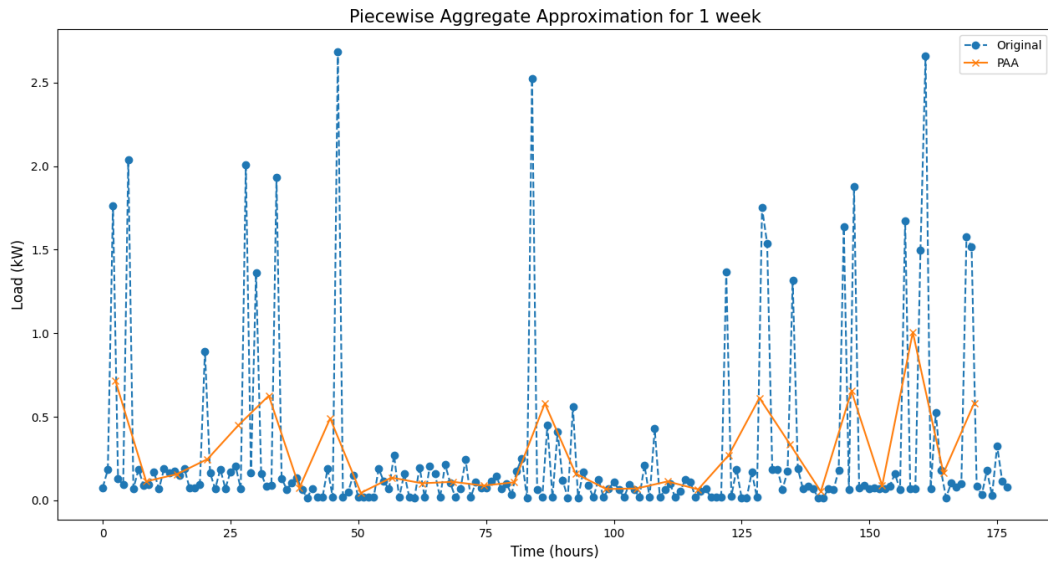


Figura 2.3.1: Esempio di applicazione di PAA su carico in kW

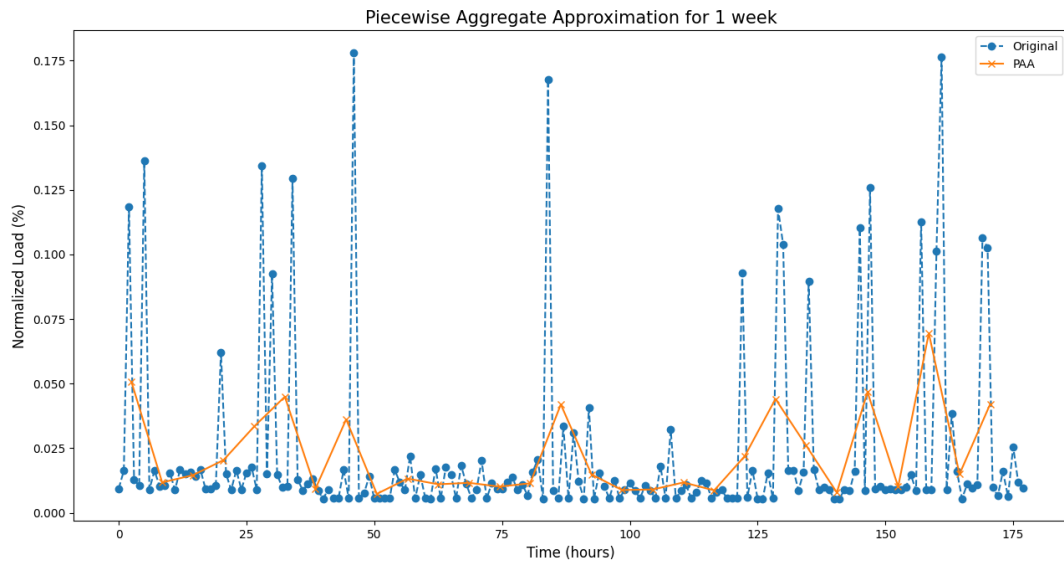


Figura 2.3.2: Esempio di applicazione di PAA su carico normalizzato

2.4 Metodi di clustering

Il clustering è una tecnica esplorativa che ci permette di estrarre informazioni da un dataset senza avere informazioni a priori su di esso. Consiste nella ricerca di strutture e pattern all'interno di un dataset, e proprio per la mancanza di etichette sui dati iniziali il clustering fa parte della cosiddetta categoria degli algoritmi non supervisionati (si parla cioè di *Unsupervised learning*)[11].

Si fa clustering ogni qualvolta si vogliono trovare gruppi (cluster, appunto) nei dati, perciò nel nostro caso i metodi di clustering ci forniscono un mezzo importante per raggruppare clienti simili così da comprendere meglio i loro modelli di consumo energetico, dandoci il vantaggio di poter lavorare con dati non etichettati.

In questo lavoro sono considerati due tipi di tecniche di apprendimento non supervisionato, il clustering K-Means ed il clustering gerarchico agglomerativo (*Agglomerative Clustering*):

Il K-means clustering è un metodo che è stato ampiamente utilizzato per il clustering del carico elettrico grazie alla relativa semplicità di implementazione. Alla base del suo funzionamento si assume che gli attributi degli oggetti da analizzare possano essere rappresentati come vettori, e che quindi formino uno spazio vettoriale.

L'obiettivo del K-Means è di ridurre al minimo la varianza totale intra-gruppo. Ciascun gruppo viene identificato tramite un centroide, o punto medio. L'algoritmo segue una procedura iterativa creando inizialmente K partizioni e assegnando i punti d'ingresso a ogni partizione o casualmente oppure utilizzando alcune informazioni euristiche, calcolando quindi il centroide di ogni gruppo. Viene costruita in seguito una nuova partizione associando ogni punto d'ingresso al gruppo il cui centroide è più vicino ad esso. Infine vengono ricalcolati i centroidi per i nuovi gruppi e così via, fino a quando l'algoritmo non converge [12].

Grazie alla sua prevalenza in letteratura, serve come un buon punto di riferimento da cui confrontare modelli più complessi, come l'Agglomerative Clustering.

L'Agglomerative è un tipo di clustering gerarchico "Bottom-Up" - cioè con un ordinamento dal basso verso l'alto - in cui si parte assegnando un cluster diverso per

ogni singolo dato in ingresso, dopodiché l'algoritmo procede gradualmente ad agglomerare più cluster insieme a due a due (sfruttando metriche opportune) fino a quando si arriva ad avere un unico cluster contenente tutti i dati o viene soddisfatto un criterio di arresto [13]. La fusione viene eseguita in base ai criteri di collegamento selezionati. I metodi dei criteri di collegamento comprendono il *Single-Linkage*, in cui la somiglianza viene valutata sulla base della coppia di elementi più vicina in due cluster; l'*Average-Linkage*, dove la somiglianza viene definita in base alla distanza media tra gli elementi dei due cluster; il *Complete-Linkage*, in cui la somiglianza è definita in base alla somiglianza dei due elementi più lontani tra i cluster; ed il *Ward-Linkage* dove la somiglianza si basa sulla minimizzazione della somma dei quadrati di qualsiasi cluster formatosi dagli attuali cluster [14]. Ai fini dell'identificazione del fotovoltaico, il numero di cluster è fissato a due, uno per i prosumer solari e uno per i consumatori, e viene utilizzato il criterio del Ward-Linkage.

Il risultato può essere meglio compreso se, mentre otteniamo i cluster, andiamo a costruire il dendrogramma. Si tratta di un grafico ad albero dove sull'asse delle ordinate è riportata la "distanza" tra i cluster e sull'asse orizzontale vengono riportati i vari dati in ingresso. In Figura 2.4.1 viene qui riportato a scopo di esempio il dendrogramma corrispondente al clustering agglomerativo con metodologia *Ward* sui dati dei consumi energetici di 41 residenti del dataset di Pecan Street (i risultati di questo clustering saranno discussi più approfonditamente nel Capitolo 3).

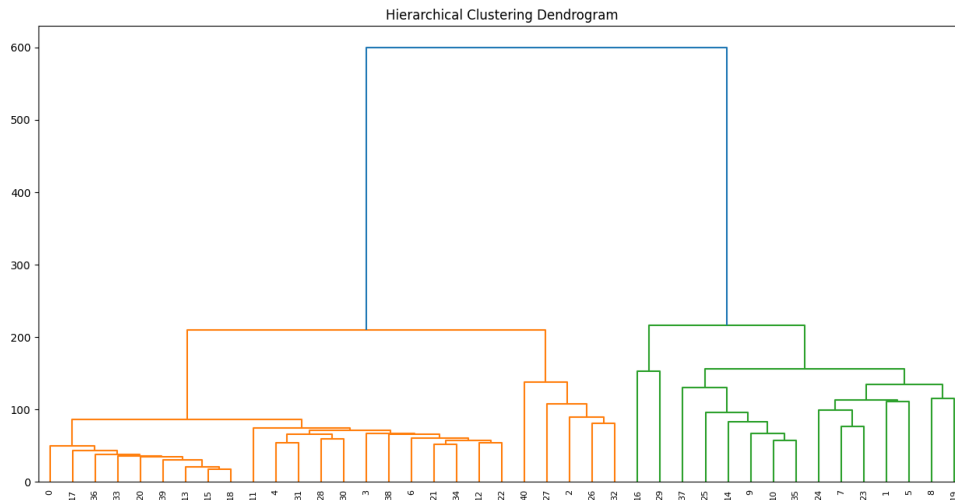


Figura 2.4.1: Esempio di dendrogramma di Clustering Agglomerativo (*Ward-Linkage*)

In questo diagramma le righe verticali corrispondono ad un cluster, quelle orizzontali ad operazioni di unione (se si usa la versione agglomerativa dell’algoritmo, che si legge dal basso verso l’alto, *bottom-up*) o di divisione (se si usa la versione divisiva dell’algoritmo, che si legge dall’alto verso il basso, *top-down*).

Il dendrogramma può essere pensato come una specie di “memoria” dei cluster, cioè che visualizza come questi cambiano in funzione della distanza massima che vogliamo applicare ai nostri dati, distanza che funzionerà come una soglia. Una volta calcolate tutte le combinazioni di cluster, è possibile modificare la soglia di clusterizzazione senza dover rieseguire nuove elaborazioni, ma semplicemente utilizzando il dendrogramma.

Questa flessibilità, in ogni caso, si paga con un costo di elaborazione più elevato di quello necessario con il K-Means, proprio perché per creare il dendrogramma è necessario elaborare tutte le combinazioni.

2.5 Etichettatura dei cluster

K-Means ed Agglomerative clustering vengono applicati in modo non supervisionato, specificando due cluster. Quindi, anche se si formano due gruppi, l'algoritmo di base non è capace di etichettare direttamente chi rappresenta i prosumer solari e chi i consumatori.

Si propone quindi un avanzamento della tecnica di clustering per automatizzare l'etichettatura dei cluster. Poiché i prosumers solari sono in grado di compensare una parte del loro consumo energetico attraverso la loro generazione in loco, per l'etichettatura dei cluster si parte dal presupposto che tutti i clienti all'interno del cluster che contiene il singolo cliente con il più basso consumo energetico debbano essere etichettati come "prosumers solari". (Va notato con attenzione che questa relazione è invertita nel caso in cui sia stata effettuata una normalizzazione dei dati a priori del clustering, poiché il carico normalizzato per un prosumer solare è superiore al carico normalizzato per un consumatore tipico).

La validità di queste ipotesi viene testata nei paragrafi 3.2 e 3.3 del capitolo seguente, eseguendo un confronto statistico del consumo semestrale dei prosumer solari e dei consumatori.

CAPITOLO 3

Esperimenti

3.1 Descrizione dei dati

Il dataset che abbiamo utilizzato per sperimentare il nostro algoritmo sono stati presi dal Pecan Street Dataport [7]. Pecan Street Inc. è un'organizzazione di ricerca e sviluppo per energia pulita e rete intelligente con sede presso l'Università del Texas, ad Austin. Svolge attività di ricerca applicata e gestisce le ricerche condotte da docenti dell'UT e da studenti e ricercatori del settore privato di aziende che sono membri del Pecan Street Consortium.

Il dataset si compone di dati raccolti tramite monitoraggio non intrusivo dagli smart meter di 73 clienti residenziali provenienti da tre diverse aree degli Stati Uniti (California, Texas, New York). Per ognuno dei residenti i dati sono stati raccolti ad intervalli di 15 minuti da più circuiti della casa, fornendo una conoscenza dettagliata di quando - e come - l'elettricità è stata utilizzata. In Figura 3.1.1 è mostrato come esempio l'andamento del consumo energetico nell'arco di sette ore di uno dei residenti registrati nel Pecan Street Dataport.

Questo particolare set di dati è inoltre unico in quanto il carico e la generazione disaggregata dei clienti sono misurati individualmente, includendo 37 impianti solari fotovoltaici e 16 veicoli elettrici.

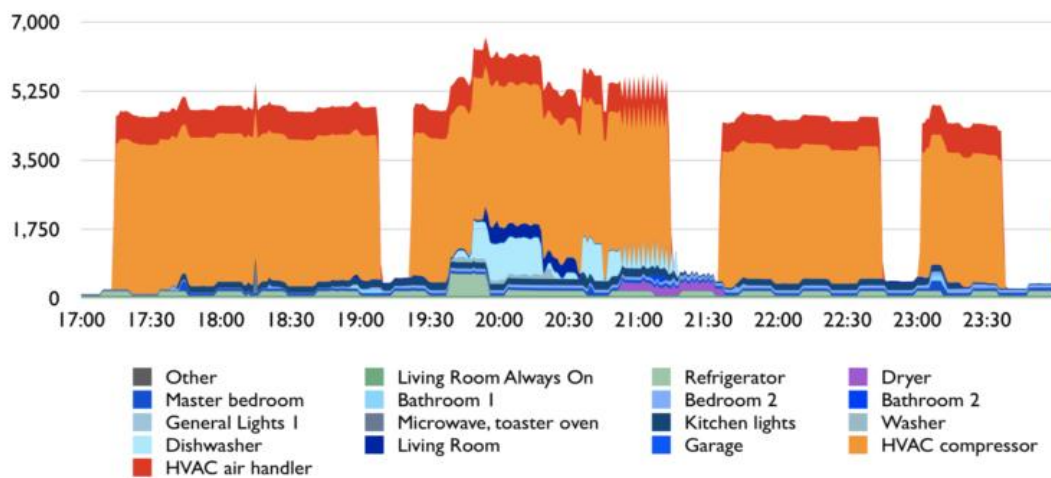


Figura 3.1.1: Uso di energia elettrica in una casa nella rete di ricerca di Pecan Street

Nonostante i dati di Pecan Street contengano anche un indicatore che mostra per ogni cliente se ha dei pannelli solari installati o meno nella sua residenza, l'approccio di identificazione proposto dal nostro algoritmo utilizza solo il consumo netto di energia dei clienti piuttosto che il carico e il fotovoltaico misurato individualmente, in quanto solo il consumo netto di energia è tipicamente disponibile per i clienti su uno schema di misurazione netto.

L'approccio è stato implementato in Python su un PC Lenovo V15-IIL a 3,2 GHz con 8 GB di memoria principale, utilizzando i pacchetti *numpy*, *pandas* e *sci-kit learn* per l'analisi dei dati, il pacchetto *minisom* per la creazione della SOM, e i pacchetti *matplotlib* e *seaborn* per la visualizzazione[16].

Il primo passo necessario per la sperimentazione dell' algoritmo è stato quello di unire i dataset di Pecan Street (originalmente tre, divisi per area geografica: California, Texas, New York) in un unico DataFrame tramite la libreria Pandas, come mostrato nella figura seguente:

```
df_cal = pd.read_csv("15minute_data_california.csv")
df_aus = pd.read_csv("15minute_data_austin.csv")
df_ny = pd.read_csv("15minute_data_newyork.csv")
dati = pd.concat([df_cal, df_aus, df_ny], ignore_index=True)
```

Figura 3.1.2: Creazione di un dataset unico (California + Austin + New York)

3.2 Pulizia dei dati

Sebbene i dati siano già stati puliti a macchina, l'analisi esplorativa dei dati ha rivelato la necessità di un'ulteriore pulizia che comprende i dati mancanti, le misurazioni errate e le linee piatte. L'approccio della Figura 2.1.2 mostrata nel Capitolo 2, è stato usato per segnalare questi clienti per l'esclusione dall'analisi.

Lavorando con dati raccolti da tre diversi stati degli USA, è stato fondamentale convertire l'intero dataframe in UTC (*Universal Time Coordinated*) per evitare problemi legati ai diversi fusi orari. Come risultato di ciò i valori temporali avranno un offset che varierà dal UTC-7:00 ore per la California, ad un UTC-4:00 ore per New York. Per replicare meglio i dati che potrebbero essere prontamente disponibili per le utility, per questo esperimento è stata utilizzata la risoluzione temporale oraria.

Come *Period Of Interest* (POI) si è deciso di prendere in esame i sei mesi da Luglio a Dicembre senza considerare l'anno di riferimento. Questa scelta particolare è stata dettata dalla necessità di mantenere il gruppo di residenti su cui viene testato l'algoritmo il più numeroso possibile. I sei mesi considerati costituiscono un intervallo temporale di un totale di 4416 ore. Per operare il cleaning dei dati nel periodo di interesse vengono rimossi dal dataframe tutti gli utenti che non possiedono dati nel semestre luglio-dicembre, ma il filtraggio del POI non si limita a questo: capita che alcuni utenti, avendo dati raccolti per due o tre anni consecutivi, possano avere più di un semestre luglio-dicembre nei dati. In questo caso, grazie al procedimento illustrato in Appendice A, viene selezionato solamente il semestre con più dati e gli altri vengono eliminati.

In uscita dal cleansing ci rimarranno solamente 43 dei 73 residenti iniziali. Il numero di clienti con misurazioni valide nel Period of Interest è mostrato nella tabella in Figura 3.2.2.

Clients total in the dataset:	73
Clients remaining after the removal of those with missing data:	46
Clients remaining after the removal of those with anomalous data:	45
Clients remaining after the removal of those with missing data:	41

Figura 3.2.2: Clienti unici rimanenti dopo la pulizia dei dati.

3.3 Risultati del clustering

A seguito della pulizia dei dati sul consumo netto di energia, la normalizzazione dei dati è stata effettuata come indicato nel paragrafo 2.3. Il clustering è stato effettuato su base semestrale per ciascuno dei due metodi discussi nel paragrafo 2.4, sia sui dati di input grezzi (in kW) che su quelli normalizzati (%).

I dati utilizzati per il clustering sono racchiusi in una matrice $C \times 4416$ dove "C" è il numero di clienti con dati completi per i mesi indicati tra luglio e dicembre, e 4416 sono le ore totali nell'arco di quei sei mesi. Ogni elemento della matrice rappresenta il valore del consumo netto di energia per quell'ora per un cliente specifico.

Per selezionare ed ordinare i valori di consumo energetico netto orario di ciascuno degli utenti è stata definita una funzione apposita, che prende in ingresso un *datetime* ed un *integer* che rappresenta un mese indicativo di inizio (luglio = 7), e calcola a partire dall'inizio di quel mese fino al *datetime* in input quante sono le ore totali. La struttura della funzione è mostrata in Figura 3.3.1:

```

def extract_hour_from_date(datatime, initial_month):
    initial_month = 7
    months = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
    hours = 0
    days = 0
    if datatime.year % 4 == 0:
        months[2] = 29
    if datatime.month != initial_month:
        for i in range(initial_month-1, datatime.month-1):
            days += months[i]
    hours = ((days + datatime.day - 1)*24) + datatime.hour
    return hours

```

Figura 3.3.1: extract_hour_from_date function

Le prime venti righe della matrice che viene data in ingresso al clustering sono invece mostrate in Figura 3.3.2: i numeri nella prima colonna da sinistra rappresentano le identità digitali dei 41 clienti rimasti dopo la fase di data clearing. Come si può vedere le ID sono state arbitrariamente ordinate per numero crescente:

	1	2	3	4	5	...	4412	4413	4414	4415
203.0	0.175	0.164	0.176	0.096	0.099	...	0.099	0.102	0.016	0.016
661.0	1.315	2.745	3.125	2.530	1.308	...	3.033	1.549	1.519	3.854
1450.0	1.946	2.157	0.430	2.478	0.908	...	1.256	0.368	0.707	0.230
1524.0	0.126	0.051	3.625	0.115	1.551	...	0.179	0.118	0.350	0.584
1731.0	0.113	0.073	0.073	0.092	0.105	...	0.440	0.440	0.554	0.552
2361.0	4.042	3.613	3.203	3.605	0.061	...	2.573	1.413	1.480	0.279
2606.0	0.075	0.076	1.694	0.241	2.780	...	0.070	0.069	0.069	0.069
2818.0	3.818	3.550	3.601	3.491	3.318	...	2.467	1.650	1.815	0.689
3039.0	0.650	0.369	0.253	0.388	0.363	...	2.877	1.740	1.887	0.568
3456.0	1.517	1.367	2.004	1.736	1.626	...	2.914	3.588	1.727	1.784
3538.0	3.156	2.541	1.328	0.412	1.320	...	1.981	1.320	1.012	0.312
3687.0	0.284	0.286	0.296	0.714	0.683	...	0.632	0.271	0.259	0.136
3864.0	2.091	0.957	0.698	2.093	0.371	...	0.078	0.132	0.057	0.057
3938.0	0.004	0.138	0.003	0.136	0.003	...	0.137	0.801	0.145	0.155
4031.0	3.967	0.701	0.655	0.668	1.188	...	3.107	2.123	2.760	1.741
4495.0	0.169	0.125	0.003	0.003	0.003	...	0.003	0.003	0.219	0.135
4767.0	3.310	0.245	1.484	1.834	0.136	...	0.000	0.000	0.000	0.000
4934.0	0.000	0.000	0.000	0.000	0.030	...	0.130	0.026	0.056	0.125
5938.0	0.067	0.061	0.061	0.145	0.294	...	0.191	0.228	0.300	0.701
6139.0	2.743	5.986	6.670	5.450	0.388	...	2.033	4.939	0.968	2.515

Figura 3.3.2: Matrice per il clustering (prime 20 righe)

Per l'implementazione del K-Means clustering, il numero di iterazioni è stato fissato a 300 e i centroidi vengono reinizializzati 10 volte. Per il clustering agglomerativo è stato utilizzato il *Ward-Linkage*, il cui funzionamento è stato descritto nel paragrafo 2.4.

3.4 Selezione del modello di classificazione

Dopo il clustering, l'algoritmo etichetta ogni cluster come "Solar Prosumer" o "Consumer" sulla base delle ipotesi fatte nel paragrafo 2.4, considerando determinati periodi di analisi del consumo energetico. La metodologia operativa del labeling automatico è illustrata nel dettaglio in Appendice B.

I grafici in Figura 3.4.1 (a-b) mettono a paragone il consumo energetico netto nell'arco dei sei mesi di POI (in kilowatt) dei prosumer solari con quello dei clienti consumatori. Nello specifico, il primo grafico fa riferimento ai risultati del labeling a valle del K-Means, mentre il secondo fa riferimento a quelli del labeling relativo all'Agglomerative Clustering.

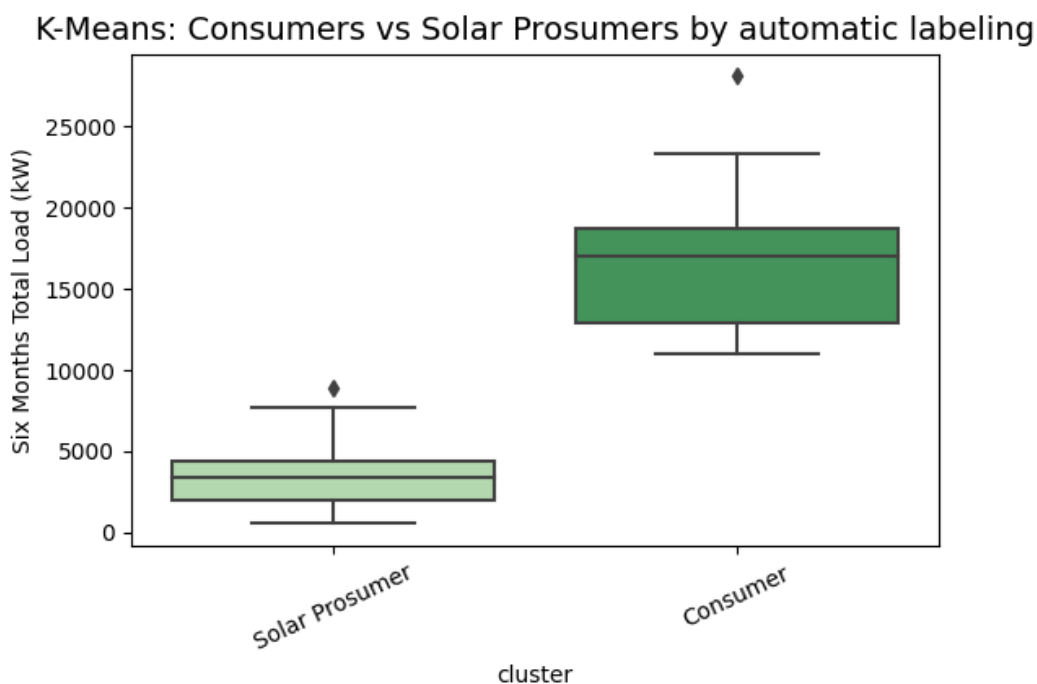


Figura 3.4.1 (a): Confronto di consumo energetico fra i *Solar Prosumer* ed i *Consumer* secondo il labelling automatico successivo al clustering K-Means

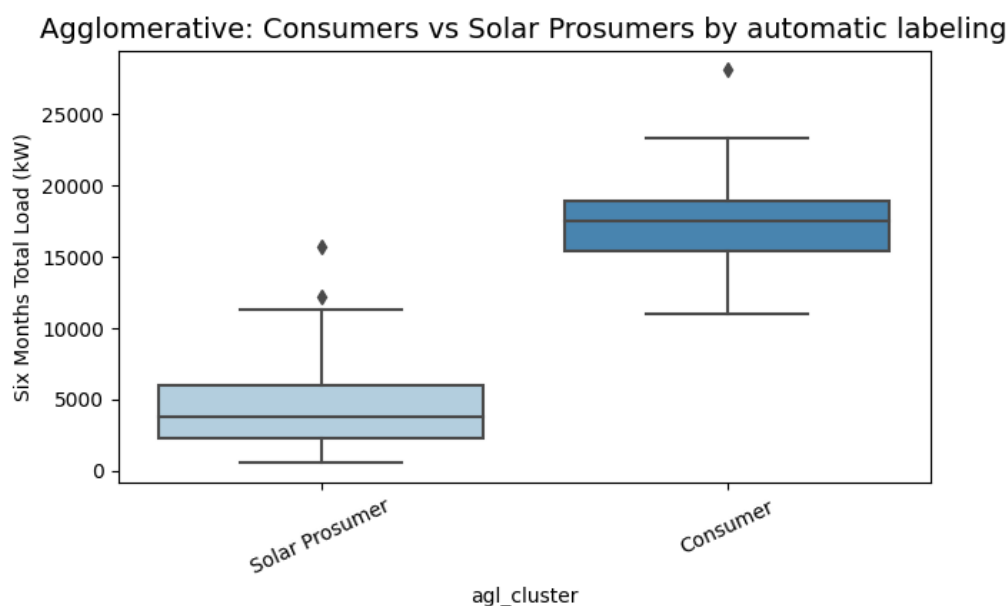


Figura 3.4.1 (b): Confronto di consumo energetico fra i *Solar Prosumer* ed i *Consumer* secondo il labelling automatico successivo al clustering gerarchico agglomerativo

Successivamente all’etichettatura dei clienti all’interno di ogni cluster, la capacità di ciascuno dei due metodi di clustering (il K-Means e l’Agglomerative) nel dividere il set di dati in proumers solari e consumatori viene valutata rispetto alle vere etichette dei clienti sulla base di due metriche distinte: Accuracy ed F1 Score.

Una volta costruito un modello di classificazione, è necessario valutare quanto siano accurate le previsioni fatte da quel modello [15]. Per quantificare questa accuratezza esistono alcune metriche di performance che ci aiutano a migliorare i nostri modelli. Possiamo esplorare le differenze tra queste metriche per un problema di classificazione binaria come il nostro:

Per prima cosa consideriamo la seguente matrice di confusione per un problema di classificazione che, a mo’ di esempio, prevede se un cliente della rete è un prosumer solare o meno:

		Predicted	
		Solar Prosumer = Yes	Solar Prosumer = No
Actual	Solar Prosumer = Yes	True Positive (TP)	False Negative (FN)
	Solar Prosumer = No	False Positive (FP)	True Negative (TN)

Figura 2.5.1: Matrice di confusione per un problema di classificazione binario

Di seguito sono riportate le metriche fondamentali per i dati di cui sopra. Nelle formule TP sta ad indicare i True Positive, TN i True Negative, FP i False Positive ed FN i False Negative:

Precision: è implicita come misura dei casi positivi correttamente identificati tra tutti i casi positivi previsti. Quindi, è utile quando i costi dei falsi positivi sono elevati.

$$Precision = \frac{TP}{TP + FP}$$

Recall: è la misura dei casi positivi correttamente identificati tra tutti i casi positivi reali. È importante quando il costo dei falsi negativi è elevato.

$$Recall = \frac{TP}{TP + FN}$$

Accuracy: è la misura di tutti i casi correttamente identificati. È più utilizzata quando tutte le classi sono ugualmente importanti e ci fornisce una visione d'insieme delle prestazioni complessive dell'algorithmo di classificazione

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

F1 Score: è la media armonica di Precision e Recall e fornisce una misura migliore dei casi erroneamente classificati rispetto alla metrica di Accuracy.

$$F1_Score = \frac{2 \times TP}{2 \times TP + FN + FP} \times 100\%$$

Riassumendo, la Accuracy viene utilizzata quando i True Positive e i True negative sono più importanti, mentre il punteggio F1 viene utilizzato quando i False Negative e i False Positive sono cruciali.

L'utilizzo della Accuracy risulta conveniente quando la distribuzione delle classi è simile, mentre il F1 Score è una metrica migliore nel caso in cui ci siano classi sbilanciate (cioè un numero limitato di prosumatori o consumatori solari). Poiché nella maggior parte dei problemi di classificazione capita che la distribuzione delle classi sia squilibrata, è importante che oltre all'Accuracy venga utilizzato anche il F1-Score come metrica secondaria per avere una validazione più obiettiva del clustering .

Per convalidare le ipotesi di etichettatura dei cluster, si confrontano i risultati in uscita dal labeling automatico con un array contenente le vere etichettature degli utenti. Per risalire alle etichettature veritiere degli utenti si procede, come mostrato in dettaglio in Appendice C, isolando le ID dei clienti che registrano una generazione energetica superiore ad una soglia minima nell'arco di tempo del POI. Nella figura seguente viene dunque mostrato il grafico di confronto di consumo energetico fra Solar Prosumer e Consumers relativo al labeling manuale:

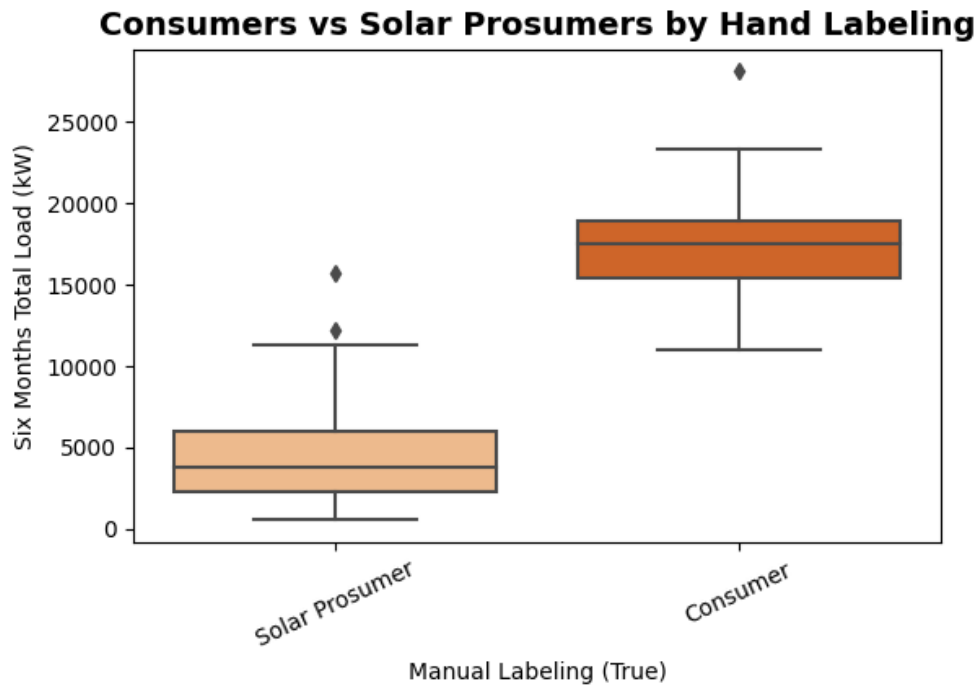


Figura 3.4.2: Confronto di consumo energetico fra i *Solar Prosumer* ed i *Consumer* secondo il labelling manuale

Secondo le ovvie aspettative, in tutti e tre i grafici di confronto energetico il consumo semestrale dei Solar Prosumer è nettamente inferiore a quello dei Consumer. Va detto però che se in questo testo avessimo riportato i grafici relativi al consumo netto normalizzato anziché in kilowatt questa relazione sarebbe stata invertita.

La valutazione delle due tecniche di clustering è stata dunque effettuata utilizzando i dati di Pecan Street. L'*Accuracy* della classificazione risultante e le altre metriche di validazione sono fornite nella Figura 3.4.3. I risultati indicano che il clustering agglomerativo è superiore al K-Means sia in base all'*Accuracy* che al F1 Score.

Risultati validazione del K-Means clustering

	precision	recall	f1-score	support
Consumer	0.83	1.00	0.91	15
Solar Prosumer	1.00	0.88	0.94	26
accuracy			0.93	41
macro avg	0.92	0.94	0.92	41
weighted avg	0.94	0.93	0.93	41

Risultati validazione dell'Agglomerative clustering

	precision	recall	f1-score	support
Consumer	1.00	1.00	1.00	15
Solar Prosumer	1.00	1.00	1.00	26
accuracy			1.00	41
macro avg	1.00	1.00	1.00	41
weighted avg	1.00	1.00	1.00	41

Figura 3.4.3 Risultati delle metriche di validazione dei clustering

L'Accuracy e il F1 Score, nel caso dell'Agglomerative clustering, raggiunge un punteggio del 100% contro un 93% per il K-Means. Va detto che punteggio ottimale dell'Agglomerative è dovuto in parte alle dimensioni ridotte del dataset utilizzato.

Con questa validazione è stato dimostrato che i prosumer solari possono essere correttamente identificati attraverso l'approccio di classificazione proposto con il clustering agglomerativo avanzato, utilizzando esclusivamente i loro dati sul consumo netto di energia.

CAPITOLO 4

Conclusioni e possibili applicazioni dell'algoritmo

In questo capitolo conclusivo sono svolte le considerazioni finali sull'elaborato di tesi, concentrando l'attenzione anche sui possibili utilizzi futuri del progetto realizzato.

Il presente lavoro di tesi si pone l'obiettivo di definire ed implementare un algoritmo machine learning non supervisionato capace di identificare i clienti generatori di energia fotovoltaica utilizzando i dati di consumo energetico netto forniti degli smart meter.

Nell'ambito della presente tesi, è stata effettuata un'attività di analisi dei consumi energetici di un quartiere residenziale allo scopo di identificare quali fra i residenti fossero produttori di energia fotovoltaica. Con l'algoritmo creato durante l'attività di tirocinio in preparazione alla tesi, si è centrato l'obiettivo principale: i clienti generatori di energia vengono identificati utilizzando unicamente i loro dati di consumo energetico netto forniti degli smart meter. L'approccio di classificazione proposto per l'identificazione basato sul clustering agglomerativo ha dimostrato un'accuratezza di classificazione molto significativa e superiore alle prestazioni del clustering K-Mean.

Il progetto realizzato apre la strada a numerosi possibili utilizzi futuri, in quanto l'algoritmo può essere applicato direttamente in tre diversi campi da parte dei pianificatori e dei gestori della rete:

- **Protezione e regolazione della tensione:** la conoscenza della posizione dei prosumer solari esistenti è fondamentale per impostare adeguatamente la protezione per ogni alimentatore, compresi i fusibili, i recloser automatici e i relè di sovracorrente. Man mano che un numero maggiore di prosumer solari si connette, la conoscenza della loro ubicazione può aiutare gli ingegneri a perfezionare o modificare le impostazioni di protezione per limitare il rischio di guasti o malfunzionamenti delle apparecchiature e i rischi associati alla sicurezza pubblica.
- **Aumento della consapevolezza operativa:** Sapere dove si trovano i prosumers solari permette agli operatori di rendere conto della presenza del fotovoltaico su un particolare alimentatore quando si esegue la commutazione, la riconfigurazione del circuito o il ripristino. La consapevolezza del fotovoltaico durante queste attività è importante in quanto la presenza del fotovoltaico può mascherare il carico reale di un determinato gruppo di clienti e il profilo di carico dei prosumers solari è probabilmente sostanzialmente diverso.
- **Rilevamento dell'anomalia:** Oltre al raggruppamento ai fini di l'identificazione dei prosumer solari, la valutazione dei clienti che commutano i cluster nel periodo di tempo valutato può rivelare anomalie nel carico del cliente, compresi lunghi periodi di assenza di generazione e di funzionamento parziale, migliorando la consapevolezza operativa.

Appendice A:

Come descritto nel paragrafo 3.2, il processo in figura A1 è stato utilizzato per filtrare i dati all'interno del Period Of Interest, facendo attenzione a non selezionare più di un semestre per ciascuna id di utente:

```
# Il Period Of Interest saranno i mesi da luglio a dicembre (anche per anni diversi)
months_series = df2['tempi'].apply(lambda x: x.month)
year_series = df2['tempi'].apply(lambda x: x.year)
first_year = df2.iloc[0]['tempi'].year
initial_month = 7 #luglio
last_month = 12 #dicembre
df2 = df2.loc[(months_series >= initial_month) & (months_series <= last_month)].dropna(subset=['tempi'])

# Creo una lista contenente i vari id dei residenti
dataid_unique = df2['dataid'].unique()

# Per ogni utente, seleziono l'anno con più dati ed elimino gli altri
for dataid in dataid_unique:
    data_years = (df2.loc[df2['dataid'] == dataid])['tempi'].apply(lambda x: x.year)
    data_years_unique = data_years.unique()
    if len(data_years_unique > 1):
        select_year = 0
        prec_num = 0
        for year in data_years_unique:
            num = len((df2.loc[df2['dataid'] == dataid]).loc[(data_years == year)])
            if num >= prec_num:
                select_year = year
                prec_num = num
        df2.loc[(df2['dataid'] == dataid) & (year_series != select_year)] = np.nan
```

Figura A.1

Appendice B:

Come descritto nel paragrafo 2.4, il clustering K-Means viene applicato in un modo non supervisionato, specificando due soli cluster. Anche se si formano due gruppi, l'algoritmo di base non è in grado di etichettare direttamente chi rappresenta i prosumer solari e chi i consumatori.

Si propone quindi un avanzamento della tecnica di clustering per automatizzare l'etichettatura dei cluster:

Poiché i prosumers solari sono in grado di compensare una parte del loro consumo energetico attraverso la loro generazione in loco, per l'etichettatura dei cluster si parte dal presupposto che tutti i clienti all'interno del cluster che contiene il singolo cliente con il più basso consumo energetico debbano essere etichettati come prosumers solari:

```
# Automatic labeling per i risultati del k-means
if matrix_user_hour['cons_tot'].idxmin() in cluster_1.index.values.tolist():
    matrix_user_hour['cluster'] = matrix_user_hour['cluster'].replace(1, "Solar Prosumer")
    matrix_user_hour['cluster'] = matrix_user_hour['cluster'].replace(0, "Consumer")

else:
    matrix_user_hour['cluster'] = matrix_user_hour['cluster'].replace(0, "Solar Prosumer")
    matrix_user_hour['cluster'] = matrix_user_hour['cluster'].replace(1, "Consumer")

# Automatic labeling per i risultati dell'agglomerative
if matrix_user_hour['cons_tot'].idxmin() in cluster_1_agl.index.values.tolist():
    matrix_user_hour['agl_cluster'] = matrix_user_hour['agl_cluster'].replace(1, "Solar Prosumer")
    matrix_user_hour['agl_cluster'] = matrix_user_hour['agl_cluster'].replace(0, "Consumer")

else:
    matrix_user_hour['agl_cluster'] = matrix_user_hour['agl_cluster'].replace(0, "Solar Prosumer")
    matrix_user_hour['agl_cluster'] = matrix_user_hour['agl_cluster'].replace(1, "Consumer")
```

Figura B.1

Appendice C:

Come descritto nel paragrafo 3.4, il processo in figura C1 è stato utilizzato per risalire alle etichettature autentiche dei clienti registrati nel dataset. Il procedimento isola i clienti con una generazione energetica semestrale superiore ad un valore minimo di soglia minima (50 kW). Le etichette dei clienti, così ricavate, vengono poi raccolte in un array utile per il confronto con quelli in uscita dai clustering di modo da svolgerne la validazione secondo le metriche definite nel paragrafo 2.4.

```
# Scriviamo y_true basandoci esclusivamente sull'energia generata. Di conseguenza avremo:  
# Se Generated Energy != 0 ----> Solar Prosumer  
# Se Generated Energy == 0 ----> Consumer  
  
matrix_user_hour['Manual Labeling (True)'] = \  
    np.array((df2.groupby("dataid").sum())\  
            ['Generated Energy'].apply(lambda x : "Consumer" if x >= 50 else "Solar Prosumer"))
```

Figura C.1

Riferimenti

- [1] L.Schirru. “Incentivi Fotovoltaico 2020: la spinta per il settore delle rinnovabili.” Mrkilowatt.com
<https://www.mrkilowatt.it/impianti-fotovoltaici/quali-incentivi-puoi-sfruttare-in-italia-per-acquistare-il-tuo-impianto-fotovoltaico-sunpower/#:~:text=Con%20l'approvazione%20del%20decreto,fin%20al%2031%20dicem%20bre%202021> (accessed Oct. 10, 2020)
- [2] A.Agrillo, V.Surace, P.Liberatore “Solare Fotovoltaico – Rapporto Statistico 2019” gse.it (accessed Oct. 10, 2020)
- [3] General Secretariat of the European Council. “2030 climate & energy frame work.” ec.europa.eu
https://ec.europa.eu/clima/policies/strategies/2030_en#tab-0-0 (accessed Oct. 10, 2020)
- [4] Zhang X, Grijalva S. A data-driven approach for detection and estimation of residential PV installations. IEEE Trans Smart Grid 2016;7(5):2477–85.
- [5] O.Zinaman, T.Bowen, and A.Aznar. An Overview Of Behind-The-meter Solar-Plus-Storage Regulatory Design. National Renewable Energy Laboratory, March 2020.
<https://www.nrel.gov/docs/fy20osti/75283.pdf>
- [6] F. Di Napoli, High Granularity Approaches For Effective Energy Delivery From Photovoltaic Sources. Università degli studi di Napoli "Federico II", A.A. 2015-2016
- [7] ‘Pecan Street Inc., Dataport 2019’, <<https://www.pecanstreet.org/about/>> [accessed September 2020].
- [8] J.Goldberg. “Normalization – Why normalize?” Codecademy.com
<https://www.codecademy.com/articles/normalization#:~:text=Min%2Dmax%20normalization%20is%20one,decimal%20between%200%20and%201.&text=That%20data%20is%20just%20as%20squished%20as%20before> (accessed Oct. 10, 2020)
- [9] Keogh E, Chakrabarti K, Pazzani M, Mehrotra S. Dimensionality reduction for fast similarity search in large time series databases. Knowl Inf Syst 2001;3(3):263–86.
- [10] V.Krishnamoorthy. “Piecewise Aggregate Approximation” vignesh
<https://vignesh.sh/posts/piecewise-aggregate-approx/> (accessed Sept. 8, 2020)
- [11] A.Vaccaro. “Come utilizzare il k-means per clusterizzare i dati con Python e Scikit-Learn”. AlessioVaccaro.com 2017 <https://www.alessioVaccaro.com/resources/kmeans.php> (accessed Sept. 8, 2020)

[12] K. Tekomo. “Numerical Examples of K-Means Clustering.” people.revoledu.com
<https://people.revoledu.com/kardi/tutorial/kMean/NumericalExample.htm> (accessed Oct. 1, 2020)

[13] C.Casadei. “Clustering gerarchico.” developersmaggioli.it
<https://www.developersmaggioli.it/blog/clustering-gerarchico/> (accessed Oct. 8, 2020)

[14] Vijaya, S.Sharma, N Batra. Comparative Study of Single Linkage, Complete Linkage, and Ward Method of Agglomerative Clustering. Publisher: IEEE, 2019

[15] P.Huilgol. “Accuracy vs. F1-Score.” Medium.com
<https://medium.com/analytics-vidhya/accuracy-vs-f1-score-6258237beca2#:~:text=Accuracy%20is%20used%20when%20the,as%20in%20the%20above%20case> (accessed Oct. 16, 2020)

[16] L.Govoni “Python e le librerie principali per il machine learning.” lorenzogovoni.com
<https://lorenzogovoni.com/python-librerie-machine-learning/> (accessed Oct. 16, 2020)

Elenco delle figure

Figura 1.1.1: Gli impianti installati in Italia tra il 2008 ed il 2019

Figura 1.1.2: Profili di autoconsumo per settore di attività nel 2019

Figura 2.1: Approccio di classificazione per l'identificazione dei prosumer solari

Figura 2.1.2: Processo di data cleaning

Figura 2.3.1: Esempio di applicazione di PAA su carico in kW

Figura 2.3.2: Esempio di applicazione di PAA su carico normalizzato

Figura 2.4.1: Esempio di dendrogramma di Clustering Agglomerativo (Ward-Linkage)

Figura 2.5.1: Matrici di confusione per un problema di classificazione binario

Figura 3.1.1: Uso di energia elettrica in una casa nella rete di ricerca di Pecan Street

Figura 3.1.2: Creazione di un dataset unico (California + Austin + New York)

Figura 3.2.2: Clienti unici rimanenti dopo la pulizia dei dati.

Figura 3.3.1: `extract_hour_from_date` function

Figura 3.3.2: Matrici per il clustering (prime 20 righe)

Figura 3.4.1 (a): Confronto di consumo energetico fra i Solar Prosumer ed i Consumer secondo il labelling automatico successivo al clustering K-Means

Figura 3.4.1 (b): Confronto di consumo energetico fra i Solar Prosumer ed i Consumer secondo il labelling automatico successivo al clustering gerarchico agglomerativo

Figura 3.4.2: Confronto di consumo energetico fra i Solar Prosumer ed i Consumer secondo il labelling manuale

Figura 3.4.3 Risultati delle metriche di validazione dei clustering

Ringraziamenti

A conclusione di questo elaborato, desidero spendere le mie più sincere parole per ringraziare tutti coloro che mi hanno aiutato e supportato durante la realizzazione dello stesso e, in generale, durante questo mio percorso universitario, giunto ormai al termine.

Un ringraziamento particolare va al professore Emanuele Principi, che è stato il mio punto di riferimento in quest'ultimo periodo. Il professore, infatti, nonostante i tempi estremamente brevi sembrassero non consentirlo, ha accettato di essere il mio tutor durante il tirocinio che ho svolto in modalità telematica e, successivamente, il relatore di questa tesi, permettendomi in questo modo di laurearmi durante questa sessione.

Desidero ringraziare la mia famiglia per essermi stata sempre accanto durante questi anni. In particolare, vorrei ringraziare i miei genitori, Amir e Monica, per aver reso possibile tutto questo, sostenendomi sia moralmente che economicamente; mio nonno Ezio, per l'interesse mostrato verso i miei studi e per essersi impegnato a non farmi mancare mai nulla; i miei fratelli, in particolar modo Keivan, ingegnere informatico, che mi ha insegnato come programmare in Python.