POLITECNICO DI TORINO

Collegio di Ingegneria Informatica

Corso di Laurea in Ingegneria Informatica

Tesi di Laurea Magistrale

Progettazione e sviluppo di un'applicazione web per la caratterizzazione energetica degli edifici mediante metodologie data-driven

Caso di studio: la città di Torino

Relatore:
Prof.ssa Tania Cerquitelli

Laureando:
Alessio La Monica

Correlatore:
Dott.ssa Evelina Di Corso

Tutore aziendale
Edison Spa
Ing. Silvia Casagrande

Anno accademico 2018-2019

Ringraziamenti

Un ringraziamento speciale va alla Prof.ssa Tania Cerquitelli per avermi supportato e seguito in maniera costante durante lo sviluppo di questa tesi.

Questo lavoro è stato svolto presso le Officine Edison, vorrei pertanto ringraziare l'Ing. Silvia Casagrande per l'accoglienza, la gentilezza e la disponibilità datami, Diego Albergo per le conversazioni spensierate e la grande simpatia, e tutti gli altri colleghi per il supporto e la gentilezza ricevuta.

Inoltre, ringrazio calorosamente la Dott.ssa Evelina Di Corso per il tempo dedicatomi, per la simpatia, il sostegno e per essersi rivelata una persona di grande aiuto in questi mesi di lavoro. Ringrazio anche il Dott. Daniele Mazzarelli e il Dott. Stefano Proto per i tempi passati insieme, e per i consigli e l'aiuto fornitomi.

Uno speciale ringraziamento va a Debhora, con la quale ho condiviso tanti bei momenti insieme durante questa esperienza.

Ringrazio la mia famiglia per il sostegno ricevuto in questi anni di lontananza, fondamentali nei momenti di maggiore bisogno.

Ringrazio infine gli amici e i colleghi del Politecnico per l'appoggio e l'aiuto ricevuto in questi anni meravigliosi.

Sommario

La certificazione energetica degli edifici è un sistema di valutazione delle prestazioni energetiche che ha l'obiettivo di fornire informazioni utili sulla qualità energetica degli immobili e favorire il miglioramento del rendimento energetico degli edifici. L'Attestato di Prestazione Energetica (APE) è un attestato che mediante una scala da A a G sintetizza il livello di efficienza energetica degli edifici tenendo conto di parametri quali: l'isolamento termico, la geometria dell'immobile, i tipi di impianti termici e l'eventuale presenza di sistemi di energia rinnovabile. Il Sistema Informativo sugli Attestati di Prestazione Energetica (SIAPE) si occupa di raccogliere e gestire tutti i certificati energetici a livello nazionale. Questo ha portato ad un aumento considerevole di dati disponibili che è possibile sfruttare tramite tecniche di *Data Mining* per estrarre della conoscenza utile a vari utilizzatori, tra cui la pubblica amministrazione, che può disporre così di un potente strumento di pianificazione per l'esecuzione di interventi di riqualificazione energetica, oppure il privato che può così individuare le aree di maggior interesse dal punto di visto dei consumi energetici dove poter comprare o affittare casa.
L'obiettivo di questo lavoro di tesi è stato quello di sviluppare un *framework* in Python che permetta di estrarre della conoscenza nascosta a partire da una grande quantità di dati relativi a certificati energetici di edifici situati nella città di Torino, e di presentarla attraverso una *dashboard* dinamica, navigabile all'interno di un'applicazione web. Un ulteriore passo raggiunto è stato non solo quello di caratterizzare la conoscenza estratta, ma anche la sua generalizzazione a certificati nuovi non presenti nel *dataset* iniziale.
Il *framework*, tramite le fasi di *Data Cleaning*, *Data Selection* e *Normalization*, permette di ripulire il *dataset* e selezionare gli attributi più rilevanti per l'analisi. Durante la fase di *Exploratory Analysis* si estrae la conoscenza mediante algoritmi di *Clustering*. Il risultato ottenuto viene presentato tramite grafici e mappe geolocalizzate all'interno di un'applicazione web. La fase di generalizzazione per nuovi

certificati permette, nel caso di attributi mancanti, la loro predizione mediante tecniche di regressione, e la classificazione.

La tesi è organizzata in 4 capitoli:

Nel **Capitolo 1** viene presentato più in dettaglio il concetto di certificazione energetica, con particolare riferimento al quadro normativo, per poi spiegare gli attributi fondamenti di un certificato di prestazione energetica.
Nel **Capitolo 2** si pone attenzione all'architettura del *framework* sviluppato, descrivendo i concetti teorici presenti in esso.
Nel **Capitolo 3** vengono mostrati i risultati ottenuti dal *framework* sui certificati presenti nel *dataset*, attraverso grafici e mappe navigabili nell'applicazione web realizzata, e la generalizzazione della conoscenza a nuovi dati in ingresso.
Nel **Capitolo 4** verrà presentato un breve sommario della tesi svolta e verranno discussi gli sviluppi futuri.

Indice

Elenco delle tabelle

Elenco delle figure

Capitolo 1

La Certificazione Energetica

La certificazione energetica è un documento prodotto da un tecnico abilitato che tiene conto dell'isolamento termico, delle caratteristiche architettoniche dell'edificio, della zona climatica di appartenenza, dell'eventuale presenza di sistemi di energia rinnovabile e in generale di tutto ciò che è responsabile dei consumi energetici.
Il 40% del consumo energetico in Europa è dovuto agli edifici [1]; la certificazione energetica nasce con l'obiettivo di promuovere il rendimento energetico degli edifici, riducendo allo stesso tempo le emissioni dei gas serra per far fronte al cambiamento climatico e raggiungere gli obiettivi definiti dal protocollo di Kyoto. Ridurre i consumi è infatti una delle priorità dell'Unione Europea che verso la fine del 2006 ha fissato l'obiettivo di ridurre del 20% il consumo di energia primaria entro il 2020.

1.1 Sviluppo del quadro normativo

In questo paragrafo viene ripercorsa l'evoluzione della normativa di riferimento per la certificazione energetica, a partire dalle *Leggi 373/1976* e *Leggi 10/1991*, fino ad arrivare ai *Decreti del 26 giugno 2015*.
La legge n.373/1976, redatta per la riduzione del consumo energetico, prevede le prime restrizioni per la progettazione, l'installazione, l'esercizio e manutenzione degli impianti termici. *La legge n.10/1991*, nata come la precedente con la finalità di ridurre i consumi di energia, è la prima legge a coordinare le modalità progettuali e la gestione del sistema edificio/impianto. *Con il D.P.R. n. 412/1993* vengono introdotte due novità, la prima riguarda la classificazione del territorio nazionale in base al numero di gradi giorno (GG), mentre la seconda la categorizzazione degli

edifici per destinazione d'uso. La *Direttiva 2002 91/CE*, anche detta EPBD (*Energy Performance of Building Directive*), sollecita gli stati membri dell'Unione Europea ad attuare una serie di misure finalizzate al miglioramento dell'efficienza energetica nel settore edilizio. Nell'EPBD viene proposto il Certificato energetico, con validità massima di 10 anni, che permette ai cittadini di conoscere l'efficienza energetica dell'edificio. La *Direttiva 2002 91/CE* viene adottata in Italia con il *D.Lgs. 19 agosto 2005 n.192*. Il *D.Lgs. n.311/2006* introduce alcune modifiche al *D.Lgs. n.192* e delle linee guida per la certificazione energetica degli immobili. Inizia a decorrere da questo decreto l'Attestato di Qualificazione Energetica (AQE). Il *D.P.R. n.59/2009* si occupa di definire le norme tecniche per il calcolo delle prestazioni energetiche degli edifici e i criteri di edifici e impianti relativi a: climatizzazione estiva, preparazione di acqua calda per usi sanitari, climatizzazione invernale e illuminazione artificiale di edifici non residenziali. Il *D.M. 26 giugno 2009* lancia le linee guida nazionali per la certificazione energetica, specificando che il certificato energetico contenga i valori di riferimento a norma di legge con le classi prestazionali e indicazione sull'efficienza energetica dell'edificio. Il 2009 è anche l'anno dell'introduzione dell'ACE, l'Attestato di Certificazione Energetica, realizzato a partire dall'1 luglio 2009 in caso di compravendita e dall'1 luglio 2010 in caso di locazione onerosa. Il *D.Lgs. n. 28/2011* introduce alcune novità, tra cui l'obbligo di utilizzare fonti rinnovabili negli edifici di nuova costruzione e sottoposti a ristrutturazionni di rilievo. Con il *D.L 63/2013* si passa dall'ACE all'APE (Attestato di Prestazione Energetica), viene inoltre previsto il rilascio dell'APE da parte del professionista in forma di dichiarazione sostitutiva di atto notorio. Nel 2015 vengono redatti tre *Decreti*. Il primo riguarda le modalità per la compilazione della relazione tecnica di progetto al fine dell'applicazione dei requisiti minimi di prestazione energetica negli edifici. Il secondo è relativo all'applicazione dei motodi di calcolo delle prestazioni energetiche e definizione dei requisiti minimi degli edifici. Il terzo è concernente l'adeguamento linee guida nazionali per la certificazione energetica degli edifici [2];

1.2 AQE, ACE ed APE

L'AQE, ovvero Attestato di Qualificazione Energetica, come detto nel paragrafo precedente viene introdotto nel *D.Lgs. n.311/2006*. È redatto alla fine dei lavori di costruzione o ristrutturazione da un tecnico non strettamente estraneo alla proprietà, come il progettista o il direttore dei lavori. Viene consegnato al comune insieme alla

documentazione per il rilascio della dichiarazione di fine lavori. Successore dell'AQE è l'ACE: l'Attestato di Certificazione Energetica, che a differenza del precedente va redatto esclusivamente da un Certificatore Abilitato. Un'altra differenza è che nell'AQE non è specificata la classe energetica dell'edificio. Dal 2013 con il *D.Lgs. n. 90/2013* viene introdotto l'Attestato di Prestazione Energetica, ovvero l'APE, un documento che si distingue per i contenuti più completi e per la maggiore quantità di informazione, che consente di comprendere anche i margini di miglioramento della prestazione energetica degli edifici. L'APE suddivide la vecchia classe più performante "A" in quattro sottoclassi, dalla più bassa, l'"A1", alla più alta , l'"A4", mentre le altre classi, dalla "B" alla "G", non subiscono modifiche. Per assegnare la classe energetica ad un edificio si tengono in considerazione diverse variabili: la climatizzazione invernale ed estiva, l'acqua calda per usi sanitari, la ventilazione, l'illuminazione e l'energia richiesta da ascensori e scale mobili. La durata dell'APE è rimasta invariata rispetto allACE. Il documento ha una validità di 10 anni se non vengono effettuate ristrutturazioni che comportino variazioni delle prestazioni energetiche dell'edificio.

1.3 La certificazione energetica nella regione Piemonte

Il *dataset* analizzato nel seguente lavoro contiene certificazioni energetiche della città di Torino rilasciate grazie al supporto di CSI-Piemonte (Consorzio per il Sistema Informativo). La regione Piemonte con la *legge regionale del 28 maggio 2007, n. 13* ha adottato le norme per la certificazione energetica degli edifici definendo le prestazioni minime, la metodologia di calcolo e le modalità per il rilascio dell'Attestato di Certificazione Energetica, che deve essere prodotto nei casi di: edifici di nuova costruzione, ristrutturazione edilizia all'atto di chiusura dei lavori ai fini dell'ottenimento dell'agibilità, compravendita o locazione di intero immobile o singole unità immobiliari [3]. Nell'ambito della Rete Unitaria della Pubblica Amministrazione Regionale (RUPAR), la regione Piemonte ha realizzato il Sistema Informativo per la Certificazione Energetica degli Edifici, denominato SICEE, che conteneva l'elenco dei certificatori e degli ACE. L'accesso al SICEE permetteva diverse operazioni, tra cui: il rilascio di copie dell'attestato di certificazione energetica, l'estrazione degli ACE per attività di controllo e la compilazione e l'invio degli certificati energetici da parte del certificatore. Con la *Delibera di Giunta Regionale 21 settembre 2015, n.*

14-2119 la Regione Piemonte sostituisce il SICEE con il Sistema Informativo Regionale per la Prestazione Energetica degli Edifici (SIPEE). Il SIPEE gestisce l'elenco dei soggetti autorizzati al rilascio dell'APE, i dati inseriti negli APE e la raccolta degli Attestati di Prestazione Energetica [4].

1.4 Descrizione degli attributi principali delle certificazioni energetiche analizzate

In questo paragrafo vengono illustrati gli attributi principali delle certificazioni energetiche presenti nel *dataset* oggetto di analisi, che include gli ACE dal 2009 al 2014 e gli APE dal 2016 al 2018.
Gli attributi possono essere suddivisi nelle seguenti categorie:

- Dati catastali
- Dati tecnici generali sul fabbricato
- Dati sui rendimenti
- Dati sugli impianti

1.4.1 Dati catastali

I dati catastali di una proprietà immobiliare che identificano univocamente tale proprietà dal punto di vista del catasto sono i seguenti:

- **Foglio**: il foglio catastale identifica una porzione di territorio comunale.
  È un numero sempre presente nella visura catastale.
- **Particella**: nota anche come numero di mappa, rappresenta nell'ambito del foglio catastale una porzione di terreno o fabbricato.
- **Subalterno**: sta ad identificare la singola unità immobiliare esistente in una particella

Sono contenuti nella visura catastale, un documento rilasciato dall'Agenzia delle Entrate.
Nel caso di certificati multipli per una stessa unità immobiliare, è stato considerato il certificato che è stato caricato più recentemente.

Altri attributi di natura geografica che permettono di delineare un edificio sono:

- **Indirizzo**: indirizzo dell'unità immobiliare a cui si riferisce l'attestato energetico.
- **Numero civico**: numero civico dell'unità immobiliare considerata.
- **Comune**: il comune a cui si riferisce l'attestato energetico in esame.
- **CAP**: il codice di avviamento postale dell'unità abitativa oggetto di esame.
- **Latitudine**: è la distanza angolare misurata in gradi che rappresenta l'arco di meridiano compreso tra l'equatore e il punto considerato.
- **Longitudine**: è la distanza angolare misurata in gradi che rappresenta l'arco di equatore compreso tra il meridiano di Greenwich ed il punto considerato.

1.4.2 Dati tecnici generali

In questo sottoparagrafo vengono date informazioni tecniche sull'unità immobiliare:

- **Anno di costruzione**: rappresenta l'anno di costruzione dell'unità immobiliare.
- **Destinazione d'uso**: si intende l'insieme delle modalità e delle finalità di utilizzo di un manufatto edilizio. Alcune delle principali destinazioni d'uso sono: industriale, residenziale e commerciale.
- **Superficie utile** [m^2]: è la superficie effettivamente calpestabile dell'unità abitativa, al netto dei tramezzi e muri, sia interni che perimetrali.
- **Fattore forma** [m^-1]: misura il rapporto tra la superficie disperdente S dell'edificio e il volume lordo riscaldato V. È una variabile molto importante ai fini della riduzione delle perdite energetiche per trasmissione. Più basso è il rapporto S/V, ovvero più l'edificio è compatto, migliori sono le prestazioni energetiche dell'edificio. Edifici con forma geometrica semplice, presentano un migliore rapporto S/V rispetto ad edifici con molte sporgenze.
- **Superficie riscaldata** [m^2]: superficie che comprende la climatizzazione invernale.
- **Volume lordo riscaldato** [m^3]: è la somma dei volumi lordi di tutti i vani coinvolti dalla climatizzazione invernale.

- **Superficie disperdente totale** [m^2]: la superficie disperdente riguarda le superfici dell'edificio che confinano con l'esterno, con il terreno e con tutti gli altri ambienti non riscaldati [5].

Destinazione d'uso	**Descrizione**
E1	Edifici adibiti a residenza e assimilabili
E1(1)	Abitazioni adibite a residenza con carattere continuativo, quali abitazioni civili e rurali, collegi,conventi, case di pena, caserme
E1(2)	Abitazioni adibite a residenza con occupazione saltuaria, quali case per vacanze, fine settimana e simili
E1(3)	Edifici adibiti ad albergo, pensione ed attività similari
E2	Edifici adibiti a uffici e assimilabili pubblici o privati, indipendenti o contigui a costruzioni adibite anche ad attività industriali o artigianali, purché siano da tali costruzioni scorporabili agli effetti dell'isolamento termico
E3	Edifici adibiti a ospedali, cliniche o case di cura e assimilabili ivi compresi quelli adibiti a ricovero o cura di minori o anziani nonché le strutture protette per lassistenza ed il recupero dei tossicodipendenti e di altri soggetti affidati a servizi sociali pubblici
E4	Edifici adibiti ad attività ricreative o di culto e assimilabili
E4(1)	Quali cinema e teatri, sale di riunioni per congressi
E4(2)	Quali mostre, musei e biblioteche, luoghi di culto
E4(3)	Quali bar, ristoranti, sale da ballo
E5	Edifici adibiti ad attività commerciali e assimilabili quali negozi, magazzini di vendita allingrosso o al minuto, supermercati, esposizioni
E6	Edifici adibiti ad attività sportive
E6(1)	Piscine, saune e assimilabili
E6(2)	Palestre e assimilabili
E6(3)	Servizi di supporto alle attività sportive
E7	Edifici adibiti ad attività scolastiche a tutti i livelli e assimilabili
E8	Edifici adibiti ad attività industriali ed artigianali e assimilabili

Tabella 1.1: Categorie di destinazioni d'uso degli edifici

Per la nostra analisi sono stati considerati esclusivamente certificati di edifici appartenenti alla destinazione d'uso E1(1), ovvero unità abitative adibite a residenza con carattere continuativo. In Tabella 1.1 viene presentato un riepilogo delle varie categorie di destinazione d'uso.

Per poter introdurre i concetti di trasmittanza delle superfici opache e trasparenti, è necessario dare una spiegazione del concetto di trasmittanza termica. La **trasmittanza termica U** rappresenta il flusso di calore che attraversa una superficie unitaria sottoposta ad una differenza di temperatura pari a 1°C.
Si misura in $W/(m^2K)$, è quindi un valore che da indicazioni sulla capacità di un metro quadro di involucro di disperdere calore in presenza di una differenza di temperatura di 1°C tra interno ed esterno. Più basso è il valore della trasmittanza termica degli elementi che costituiscono l'involucro dell'edificio, minore sarà il flusso di calore che li attraversa. Quando si installano nuove porte o finestre, è importante tenere conto di questo valore e scegliere infissi con un basso valore di trasmittanza termica, così da ridurre le dispersioni di calore.

- **Trasmittanza opaca** $[W/m^2K]$: rappresenta la trasmittanza media ponderata relativa agli elementi opachi dell'edificio contigui con l'ambiente esterno.
- **Trasmittanza trasparente**$[W/m^2K]$: rappresente la trasmittanza media ponderata relativa agli elementi trasparenti dell'edificio contigui con l'ambiente esterno.
- **Zona climatica**: si tratta di aree del territorio italiano caratterizzate da uguale clima, per le quali è possibile immaginare condizioni identiche o pressochè simili. Il territorio italiano è suddiviso in sei zone climatiche, dalla A, quella più calda, alla F, la più fredda (vedi Tabella 1.2). La suddivisione dell'Italia in zone climatiche è stabilita dal *D.P.R. 26/08/1993 n. 412*. Un'area appartiene ad una zona climatica piuttosto che ad un'altra grazie al valore di un parametro, chiamato Grado Giorno (GG) [6].
- **Gradi Giorno (GG)**: i gradi giorno (GG) corrispondono alla somma, estesa a tutti i giorni dell'anno, delle sole differenze positive giornaliere tra la temperatura ambiente e la temperatura media esterna giornaliera. In Italia la temperatura ambiente convenzionale è fissata a 20°C.

Zona Climatica	Gradi Giorno	Ore Giornaliere	Periodo
A	<600	6	1/12 - 15/03
B	600-900	8	1/12 - 31/03
C	901-1400	10	15/11 - 31/03
D	1401-2100	12	15/11 - 31/03
E	2101-3000	14	15/10 - 15/04
F	>3000	-	-

Tabella 1.2: Gradi giorno e zona climatica

1.4.3 Rendimenti e indici di prestazioni

I sistemi reali di riscaldamento presentano alcune perdite di calore, è necessario che l'energia primaria fornita al corpo scaldante sia maggiore di quella che esso emana verso l'ambiente. Per questo motivo, nel considerare le prestazioni energetiche di un edificio, occorre tenere in considerazione il contributo dei rendimenti medi stagionali dei quattro sottosistemi dell'impianto: il sistema di generazione di energia termica, quello di distribuzione del calore, quello di emissione e per ultimo quello di regolazione.
Vengono di seguito descritti i rendimenti:

- **Rendimento di generazione**: il rendimento di generazione ETAG medio stagionale è il rapporto tra il calore utile prodotto dal generatore nella stagione di riscaldamento e l'energia fornita nello stesso periodo sotto forma di combustibile ed energia elettrica. Il miglioramento di tale rendimento non dipende esclusivamente da fattori costruttivi dei generatori ma dipende anche dal modello di conduzione, da scelte progettuali e dal tipo di regolazione [7].
- **Rendimento di distribuzione**: il rendimento di distribuzione ETAD è il rapporto tra la somma del calore utile emesso dai corpi scaldanti e del calore disperso dalla rete di distribuzione all'interno dell'involucro riscaldato dell'edificio ed il calore in uscita dall'impianto di produzione ed immesso nella rete di distribuzione. Tale rendimento caratterizza l'influenza della rete di distribuzione sulla perdita passiva di energia termica [7].

- **Rendimento di emissione**: il rendimento di emissione ETAE è il rapporto tra il calore richiesto per il riscaldamento degli ambienti con un sistema di emissione teorico di riferimento in grado di fornire una temperatura ambiente perfettamente uniforme ed uguale nei differenti locali ed il sistema di emissione reale. Per migliorare il rendimento di emissione di un impianto di riscaldamento, è importante passare a sistemi ad alto rendimento come i pannelli a pavimento o a parete [7].
- **Rendimento di regolazione**: il rendimento di regolazione ETAR è il rapporto tra il calore richiesto per il riscaldamento degli ambienti con una regolazione teorica perfetta ed il calore richiesto per il riscaldamento degli stessi ambienti con un sistema di regolazione reale. Può andare da un minimo di 0,88 tipico dei pannelli radianti a regolazione manuale fino a valori di circa 0,94 con i termosifoni [7].

Per avere un'indicazione riepilogativa dell'efficienza energetica di un sistema di climatizzazione invernale, si può considerare il prodotto dei quattro rendimenti energetici, che viene chiamato **ETAH**.
Ai fini della classificazione energetica di un edificio, è utile tenere in considerazione anche alcuni indici di prestazione energetica:

- $\mathbf{EP_{gl,nren}}$[kWh/m^2*anno*]: l'indice di prestazione energetica globale non rinnovabile tiene conto di tutti i servizi energetici forniti agli edifici, ovvero del fabbisogno di energia primaria non rinnovabile per la climatizzazione invernale ed estiva ($EP_{H,nren}$ e $EP_{C,nren}$), per la ventilazione ($EP_{V,nren}$), per la produzione di acqua calda sanitaria ($EP_{W,nren}$), e nel caso di settore non residenziale, per il trasporto di persone o cose ($EP_{T,nren}$) e per l'illuminazione artificiale ($EP_{L,nren}$) [8].
- $\mathbf{EP_{H,nd}}$[kWh/m^2*anno*]: esprime l'indice di prestazione termica utile per la climatizzazione invernale dell'edificio, è dato dal rapporto tra il fabbisogno di energia annuo di energia termica dell'edificio e la superficie utile [8].
- $\mathbf{EP_H}$[kWh/m^2*anno*]: l'indice di prestazione energetica per la climatizzazione invernale è dato dal rapporto tra l'$EP_{H,nd}$ e dal rendimento dell'impianto di riscaldamento [8].
- $\mathbf{EP_C}$: l'indice di prestazione energetica per la climatizzazione estiva deriva dall'indice di prestazione termica utile per la climatizzazione estiva dell'edificio ($EP_{C,nd}$) e dal rendimento dell'impianto di raffrescamento [8].

1.5 La classe energetica

La classe energetica è un indicatore sintetico del grado di efficienza energetico di un edificio, viene espressa attraverso dieci etichette, che vanno dalla A4 (la classe in cui rientrano gli edifici più efficienti) alla G (la classe che include gli edifici che consumano più energia), come mostrato in Figura 1.2.
Si ottiene confrontando l'indice di prestazione energetica globale non rinnovabile dell'edificio in oggetto con quello dell'edificio di riferimento. Di seguito una figura che mostra la classificazione degli edifici attraverso confronto con $EP_{gl,nren}$.

	Classe A4	$\leq 0{,}40\ EP_{gl,nren,rif,standard\ (2019/21)}$
$0{,}40\ EP_{gl,nren,rif,standard\ (2019/21)} <$	**Classe A3**	$\leq 0{,}60\ EP_{gl,nren,rif,standard\ (2019/21)}$
$0{,}60\ EP_{gl,nren,rif,standard\ (2019/21)} <$	**Classe A2**	$\leq 0{,}80\ EP_{gl,nren,rif,standard\ (2019/21)}$
$0{,}80\ EP_{gl,nren,rif,standard\ (2019/21)} <$	**Classe A1**	$\leq 1{,}00\ EP_{gl,nren,rif,standard\ (2019/21)}$
$1{,}00\ EP_{gl,nren,rif,standard\ (2019/21)} <$	**Classe B**	$\leq 1{,}20\ EP_{gl,nren,rif,standard\ (2019/21)}$
$1{,}20\ EP_{gl,nren,rif,standard\ (2019/21)} <$	**Classe C**	$\leq 1{,}50\ EP_{gl,nren,rif,standard\ (2019/21)}$
$1{,}50\ EP_{gl,nren,rif,standard\ (2019/21)} <$	**Classe D**	$\leq 2{,}00\ EP_{gl,nren,rif,standard\ (2019/21)}$
$2{,}00\ EP_{gl,nren,rif,standard\ (2019/21)} <$	**Classe E**	$\leq 2{,}60\ EP_{gl,nren,rif,standard\ (2019/21)}$
$2{,}60\ EP_{gl,nren,rif,standard\ (2019/21)} <$	**Classe F**	$\leq 3{,}50\ EP_{gl,nren,rif,standard\ (2019/21)}$
	Classe G	$> 3{,}50\ EP_{gl,nren,rif,standard\ (2019/21)}$

Figura 1.1: Scala di classificazione degli edifici sulla base dell'indice di prestazione energetica globale non rinnovabile $EP_{gl,nren}$ (Decreto n. 162 - 15 luglio 2015)

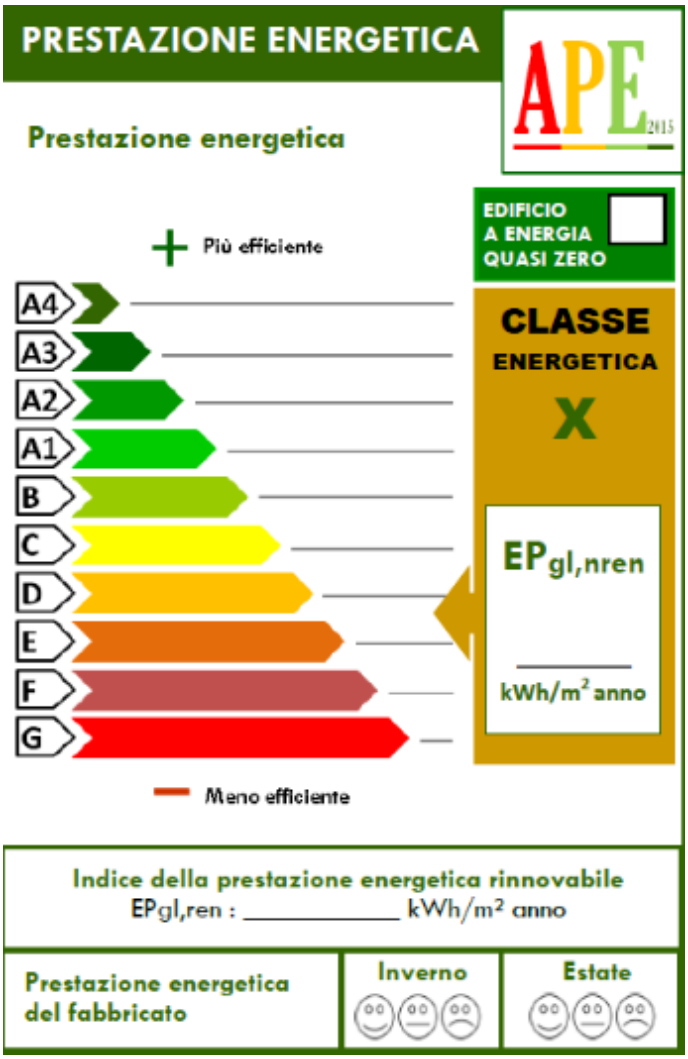

Figura 1.2: Le classi energetiche degli edifici.

Capitolo 2

Framework TUCANA

In Figura 2.1 viene mostrato il *framework* TUCANA (*Turin Certificates Analysis*), progettato e sviluppato per estrarre e visualizzare la conoscenza a partire da certificazioni energetiche relative alla città di Torino, e la predizione di valori mancanti e la successiva etichettatura di *Clustering* per nuovi dati in ingresso.

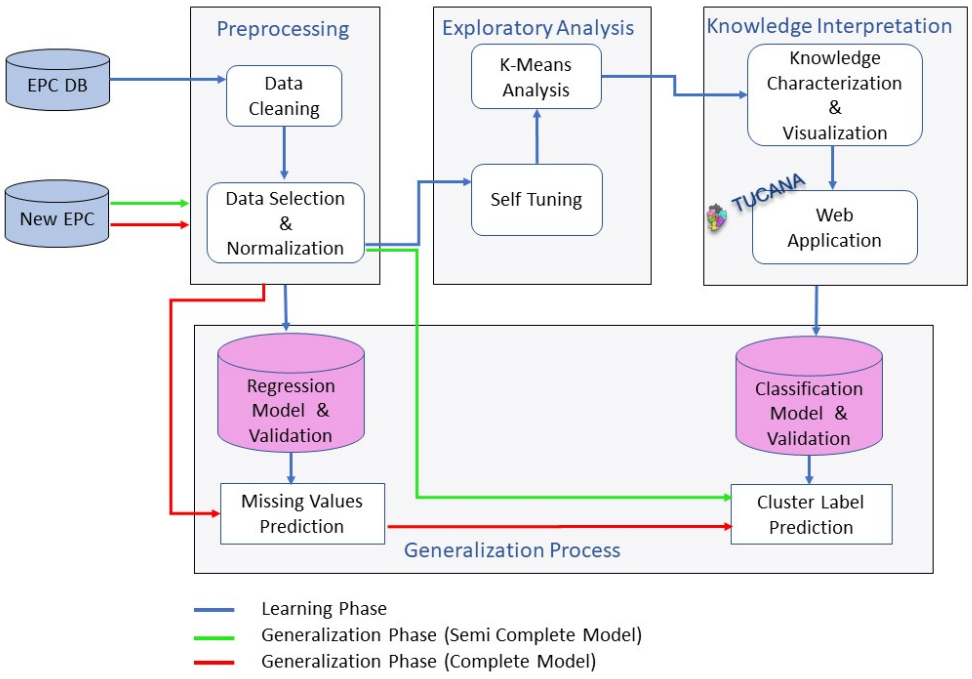

Figura 2.1: *Framework* TUCANA

Il *framework* è costituito da quattro blocchi principali.

- **Data Preprocessing**: comprende la fase di *Data Cleaning*, *Data Selection* e *Normalization*.
- **Exploratory Analysis**: include il raggruppamento in *cluster* dei dati attraverso l'algoritmo K-Means.
- **Knowledge Interpretation**: comprende la *Knowledge Characterization*, attraverso l'utilizzo di alberi di decisione, *radar chart* e *boxplot*, la *Knowledge Visualization* mediante l'utilizzo di mappe geolocalizzate interattive, e infine, l'applicazione web sviluppata per permettere la fruizione dei contenuti a più *stakeholder*.
- **Generalization Process**: fase che consente a nuovi dati in ingresso al *framework*, la predizione di valori mancanti, nel caso ce ne fossero, e la predizione dell'etichetta di *Clustering*.

2.0.1 KDD

La costante crescita della quantità dei dati prodotta quotidianamente e l'elevato aumento delle capacità di calcolo dei computer sono due fattori chiave che hanno portato ad un'urgente esigenza di sviluppo di nuovi strumenti e tecniche per assistere gli analisti all'estrazione di conoscenza utile.
Queste tecniche e strumenti sono alla base del **Knowledge Discovery in Database**, termine coniato nel 1989 per esprimere che la conoscenza è il prodotto finale di un processo *data-driven*. Il termine KDD indica l'intero processo di ricerca di nuova conoscenza dai dati, mentre il termine **Data Mining** si riferisce ad un particolare step di questo processo, ovvero all'applicazione di specifici algoritmi per estrarre *pattern* dai dati. Il KDD unifica operazioni automatiche e scelte, per estrarre conoscenza significativa da un'elevata quantità di dati eterogenei. È un processo interattivo ed iterativo che può essere descritto in nove fasi [9]:

- Sviluppo e approfondimento del dominio di applicazione, della conoscenza disponibile a priori e identificazione dell'obiettivo del processo KDD dal punto di vista dell'utente.

- Creazione di un *dataset* target: selezione del *dataset* o focalizzazione su un sottoinsieme di variabili o di campioni, sulla quale viene sviluppata la ricerca di conoscenza.

- Pulizia dei dati e *pre-processing*: ovvero operazioni base come la rimozione del rumore o degli *outlier* se necessario, raccolta di informazioni necessarie per modellare o tener conto del rumore, consolidazione di strategie per gestire dati mancanti e tempo-varianti.

- Riduzione dei dati e proiezione: rappresentazione dei dati in modo appropriato, secondo gli obiettivi della ricerca. Riduzione del numero di variabili da sottoporre al processo di ricerca.

- Identificazione dell'obiettivo del KDD: stabilire se si tratta di un problema di classificazione, regressione, *Clustering*...

- Scelta dell'algoritmo di *Data Mining*: selezione dei metodi da usare per ricercare pattern nei dati. Questo step si occupa di scegliere quali modelli e parametri potrebbero essere idonei e il *matching* di un particolare metodo di *Data Mining* con i criteri generali del KDD.

- *Data Mining*: ricerca di *pattern* di interesse in una particolare forma di rappresentazione. Il risultato di questo step è considerevolmente influenzato dalla correttezza delle fasi precedenti.

- Interpretazione dei *pattern* ottenuti ed eventuale ritorno agli step iniziali per ulteriori iterazioni. Questa fase può includere la visualizzazione dei pattern estratti.

- Consolidamento della conoscenza estratta attraverso documentazione e report alle parti interessate.

In Figura 2.2 sono mostrati alcuni step base del KDD.

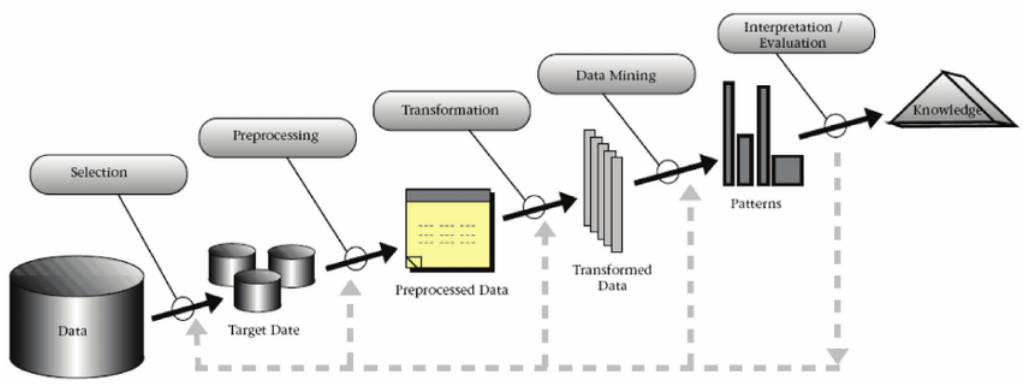

Figura 2.2: Fasi del processo di KDD secondo Fayyad, Piatetsky-Shapiro e Smyth.

2.1 Data Preprocessing

La fase di *preprocessing* è importante perchè i dati nella realtà possono essere incompleti: ovvero presentare attributi non valorizzati, non avere attributi d'interesse per gli scopi di analisi, oppure presentare solo grandezze sommarizzate. I dati possono essere anche rumorosi: ovvero presentare *outlier* ed errori e inconsistenti: presentare discrepanze nei codici e nei nomi degli attributi. A causa di questi motivi, questa fase è molto lenta, consuma infatti circa il 70% - 80% del tempo necessario al KDD. Il *preprocessing* è sicuramente fondamentale, infatti come afferma la legge **GIGO** (*Garbage in, Garbage out*), più bassa è la qualità dei dati, più scarsi saranno i risultati nella fase di *Data Mining*.
Il blocco di *Preprocessing* include al suo interno la fase di *Data Cleaning*, che verrà spiegata subito dopo aver presentato il *dataset* oggetto di analisi.

2.1.1 Dataset

Il *dataset* a disposizione raccoglie al suo interno Attestati di Certificazione Energetica, che vanno dal 2009 al 2014, e Attestati di Prestazione Energetica, compresi tra il 2016 e il primo semestre 2018, forniti dal CSI Piemonte.
I record fanno riferimento esclusivamente a certificati della città di Torino, di edifici appartenenti alla destinazione d'uso E(1). Gli attributi presenti nel *dataset* sono stati selezionati opportunamente dall'esperto di dominio e sono quelli che influiscono maggiormente sulle prestazioni energetiche degli edifici.

2.1.2 Data Cleaning

Expert-Driven univariate analysis

Questa fase del *framework* prevede l'applicazione di filtri, definiti dall'esperto di dominio, su alcuni attributi energetici ritenuti molto importanti ai fini dell'analisi, che sono suddivisi in attributi energetici (fattore forma, trasmittanza media delle superfici opache e trasparenti) e attributi che riguardano l'efficienza dei sottosistemi di riscaldamento (rendimento di generazione e distribuzione).

Address resolution

La fase di risoluzione degli indirizzi è cruciale per l'obiettivo di questo lavoro di tesi; è importante che gli indirizzi dei certificati energetici siano corretti per poterli visualizzare in mappe geolocalizzate navigabili nell'applicazione web sviluppata.
Gli attributi geospaziali: indirizzo, numero civico, latitudine, longitudine e CAP, essendo campi testuali liberi, presentano in molti casi errori di battitura, che è necessario correggere.
Il *framework* utilizzato mette a disposizione due strumenti per la risoluzione degli indirizzi, uno di questi è l'utilizzo di una particolare API (*Application Programming Interface*) messa a disposizione da Google, l'API **Geocoding** , che dato un indirizzo in ingresso, se presenti errori in esso, restituisce l'indirizzo corretto e in aggiunta il CAP e le coordinate geografiche. Il servizio fornito da Google è uno dei più precisi e affidabili ma presenta la limitazione di essere a pagamento. Per ottenere una singola risoluzione di indirizzo, il prezzo da pagare è di $0.005 fino a 100.000 richieste, per poi scendere a $0.004 se si fanno da 100.001 a 500.000 richieste [1].
Per la risoluzione della maggior parte degli indirizzi si è scelto allora di utilizzare un'alternativa, che si basa sul confronto degli indirizzi del *dataset* con quelli presenti nel viario di Torino[2], un particolare *dataset* fornito dal Geoportale della città di Torino contenente tutti gli indirizzi, ad ognuno dei quali è associato il suo CAP, coordinate e circoscrizione. Il confronto degli indirizzi si basa su un indice di similarità, calcolato a partire dalla distanza di *Levenshtein* [10], una misura per la differenza tra due stringhe, introdotta dallo scienziato russo Vladimir Levenshtein

[1] https://developers.google.com/maps/documentation/geocoding/usage-and-billing

[2] http://geoportale.comune.torino.it/web/

nel 1965. Esprime il minimo numero di modifiche elementari che permettono di trasformare una stringa A in una stringa B. Le modifiche elementari riguardano:

- La cancellazione di un carattere.
- L'inserimento di un carattere.
- La sostituzione di un carattere con un altro.

La distanza di *Levenshtein* tra due stringhe è definita come:

$$L = 2 \times NS + I + C$$

Con S definito come il numero di sostituzioni necessarie a trasformare la prima stringa nella seconda, I è il numero di inserimenti, mentre C è il numero di cancellazioni. L'indice di similarità tra due stringhe è definito come:

$$Levenshtein_{\text{ratio}} = \frac{Len_{\text{FS}} - L}{Len_{\text{FS}}}$$

Dove Len_{FS} rappresenta la somma della lunghezza tra la prima e la seconda stringa. Il primo passo della risoluzione degli indirizzi previsto dal *framework* è quello di convertire i caratteri in formato ASCII. Il passo successivo riguarda il calcolo della distanza di *Levenshtein* tra ogni coppia di indirizzi, in modo tale da ottenere l'indice di similarità, che sarà compreso tra 0 e 1, dove 1 indica la totale similarità dei due indirizzi, mentre lo 0 la totale dissimilarità.
Se l'indice di similarità è maggiore o uguale ad una certa soglia definita dall'utente, l'indirizzo presente nel *dataset* viene sostituito con quello presente nel viario di Torino, assieme al numero civico, al CAP e alle coordinate. Se invece l'indice di similarità non supera la soglia prefissata, viene utilizzato il servizio di Google Geocoding.

Outlier Detection

Quando si parla di *outlier* in statistica ci si riferisce ad un dato che è molto diverso da tutti gli altri rimanenti. Spesso gli *outlier* rappresentano del rumore nei dati, che può insorgere nella fase di raccolta dei dati [11].

I metodi di rilevamento degli *outlier* sono di fondamentale importanza per rimuoverli dalle analisi, perchè la loro presenza comprometterebbe i risultati. Il *framework* realizzato utilizza sia tecniche univariate che multivariate. Per quanto riguarda le tecniche univariate utilizzate, qui di seguito viene data una spiegazione:

- **gESD (generalized Extreme Studentized Deviate)**: è un metodo per la rilevazione di uno o più *outlier* su una distribuzione normale di dati, introdotto da Rosner nel 1983 [12]. Il limite principale del test di Grubbs e Tietjen-Moore è che bisogna specificare esattamente il numero sospetto di *outlier*, mentre il metodo di Rosner richiede di fissare un limite k superiore a tale valore. il gESD si basa sul calcolo di R_1,...,R_k statistiche, calcolati a partire dai campioni, successivamente ridotti, di dimensione *n,n-1,...,n-k+1*. Per il campione completo, si ha la statistica:

$$R_1 = \frac{max|x_1 - \bar{x}|}{s}$$

con:

$$\bar{x} = \frac{(\sum x_\mathrm{i})}{n}$$

$$s^2 = \frac{(\sum (x_\mathrm{i} - \bar{x})^2)}{(n-1)}$$

R_2 è calcolato in modo analogo dal campione di dimensione ridotta *n-1* attraverso l'eliminazione dell'osservazione che massimizza $|x_\mathrm{i} - \bar{x}|$, similmente per R_3 fino a R_k. I valori critici del test sono determinati specificando α e dopo trovando β e $\lambda(\beta)$, tale che:

$$Pr[R_\mathrm{i} > \lambda_\mathrm{i}(\beta)|H_0] = \ \beta, i = 1, ..., k$$

e:

$$Pr\left\{ \bigcup_{i=1}^{k} [R_\mathrm{i} > \lambda_\mathrm{i}(\beta) \ \middle| \ H_0] \right\} = \ \alpha$$

Se si ha che tutti gli R_i sono $\leq \lambda_{\mathrm{i}}(\beta)$, allora non si hanno *outlier*. Se invece alcuni R_i sono $> \lambda_{\mathrm{i}}(\beta)$, il metodo definisce il numero di *outlier* pari ad l, inteso come il massimo valore di i per cui $R_i > \lambda_{\mathrm{i}}(\beta)$.

- **Percentile Outlier Detection**: metodo per la cancellazione degli *outlier* fondato sul concetto di percentile [13]. Per percentile p, dato un insieme ordinati di dati, si intende quel valore che è maggiore di una percentuale p dei dati e minore della restante percentuale 100 - p, dove p è un numero nel range [0 - 100]. Questo metodo si presta bene per la valutazione della distribuzione dei dati, permettendo la rimozione di una coda dei dati.

Come tecnica multivariata, il *framework* utilizza il:

- **DBSCAN**: è un algoritmo di *Clustering* introdotto da Martin Ester, Hans-Peter Kriegel, Jörg Sander e Xiaowei Xu nel 1996 [14]. È il primo ad introdurre il concetto di densità. L'idea chiave è che ogni punto di un *cluster* deve avere nelle vicinanze di un certo raggio almeno un certo numero di altri punti, ovvero la densità nelle vicinanze del punto considerato deve superare una certa soglia. Il DBSCAN si basa su due parametri: *Epsilon e minPoints*. *Epsilon* specifica quanto i punti dovrebbero essere vicini l'uno con l'altro per essere considerati parte dello stesso *cluster*. Quindi se la distanza tra due punti è minore o uguale a *Epsilon*, allora questi due punti sono considerati vicini. *minPoints* è il minimo numero di punti che forma una regione densa.
  Quindi dato un punto p, si definisce *Eps-neighborhood* come:

$$N_{\mathrm{Eps}}(p) = \{\, q \in D \mid dist(p,q) \leq Eps \,\}$$

  Ci sono due tipi di punti in un *cluster*, i *core points* (punti dentro il *cluster*) e i *border points* (punti sul bordo del *cluster*). Un *Eps-neighborhood* di un border point contiene significativamente meno punti di un *Eps-neighborhood* di un *core point*. Un punto p è *directly density-reachable* da un altro punto q se vengono soddisfatte le seguenti condizioni:

$$p \in N_{\mathrm{Eps}}(q)$$

$$|N_{\text{Eps}}(q)| \geq MinPts$$

I *border points* per essere riconosciuti come parte di un *cluster* devono appartenere all*Eps-neighborhood* di un *core point*. Un *core point* deve avere un minimo numero di punti allinterno del suo *Eps-neighborhood* per essere considerato tale.
Se il valore scelto di *Epsilon* è troppo piccolo, una grande quantità di dati non verrà clusterizzata e verrà quindi considerata come outlier, che sono quindi punti che non sono nè *core points* nè *border points*, come mostrato nella seguente figura:

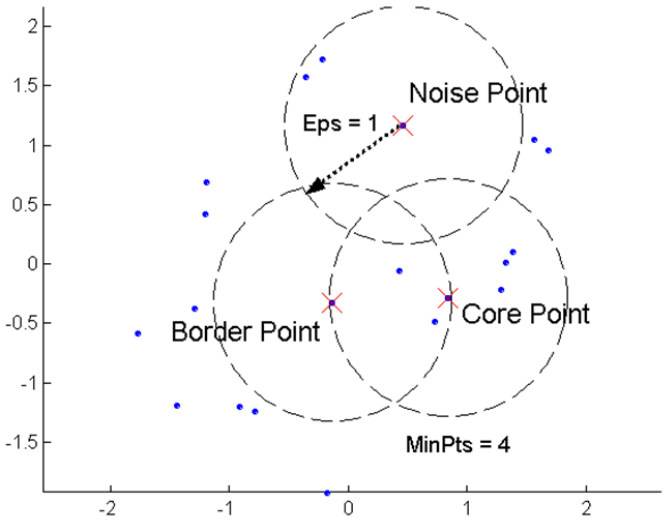

Figura 2.3: Esempio di punti core, border e noise ©Tan,Steinbach, Kumar, Introduction to Data Mining, McGraw Hill 2006

Se invece il valore di *Epsilon* è scelto troppo grande, i *cluster* si uniranno tra loro e la maggioranza di oggetti farà parte dello stesso *cluster*.
Nella figura seguente a sinistra possiamo notare come un algoritmo di *Clustering* classico, come il *K-Means*, non tenga conto della multi-dimensionalità, mentre l'immagine a destra mostra come il DBSCAN può contorcere i dati in forme e dimensioni differenti, allo scopo di trovare *cluster* simili.

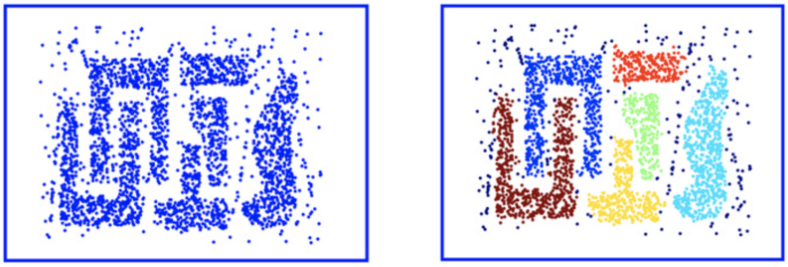

Figura 2.4: Confronto tra algoritmo di *Clustering* tradizionale con algoritmo DBSCAN

Il DBSCAN è molto buono a separare *cluster* di alta densità con *cluster* di bassa densità all'interno di un dato *dataset*, mentre non è molto adatto a gestire *cluster* di densità variabile e di simile densità.

2.1.3 Data selection and Normalization

Data selection

Questa fase del *framework*, con il supporto dell'esperto di dominio, permette di stabilire quali attributi sono rilevanti per l'analisi, sfruttando un'analisi di correlazione, che consente di misurare quanto un attributo è legato ad un altro. La correlazione degli attributi numerici viene misurata attraverso il *coefficiente di Pearson* [15] e visualizzata attraverso una matrice di correlazione.
Il *coefficiente di Pearson* viene calcolato come:

$$r = \frac{\sum_i (x_\mathrm{i} - \bar{x_\mathrm{i}}) \sum_i (y_\mathrm{i} - \bar{y_\mathrm{i}})}{\sqrt{\sum_i (x_\mathrm{i} - \bar{x_\mathrm{i}}^2)} \sqrt{\sum_i (y_\mathrm{i} - \bar{y_\mathrm{i}}^2)}}$$

Assume un valore compreso tra -1 e +1 se le variabili sono correlate, e 0 se sono scorrelate tra di loro.

Normalization

In questa sezione verranno illustrate alcune tecniche di normalizzazione, processo necessario per consentire di trasformare i dati a disposizione in un intervallo comune, ad esempio da [-1,+1] [16].

- **Min-Max**: esegue una trasformazione lineare sul dominio di partenza dei dati. Dato un attributo A e siano min_A e max_A, rispettivamente il valore minimo e massimo di A, la normalizzazione Min-Max trasforma un valore v di A in un nuovo valore v', con range di valori compreso nel nuovo intervallo [$newMin_A$, $newMax_A$].

$$v'_{\mathrm{i}} = \frac{v_{\mathrm{i}} - min_{\mathrm{A}}}{max_{\mathrm{A}} - min_{\mathrm{A}}}(newMax_{\mathrm{A}} - newMin_{\mathrm{A}}) + newMin_{\mathrm{A}}$$

  Presenta come svantaggio la difficoltà di gestire bene gli *outlier*.
- **Z-Score**: la normalizzazione Z-Score normalizza i dati di un attributo A basandosi sulla media e sulla deviazione standard. È definita nel modo seguente:

$$v'_{\mathrm{i}} = \frac{v_{\mathrm{i}} - \bar{A}}{\sigma_{\mathrm{A}}}$$

  Dove $\bar{A}$ rappresenta la media e σ_{A} la deviazione standard dell'attributo A. La normalizzazione Z-Score gestisce bene gli *outlier* e non necessita di impostare a priori il valore minimo e massimo dell'attributo A.

2.2 Exploratory Analysis

La fase di Data Preprocessing ha consentito di ottenere un *dataset* pulito, con la quale è possibile avviare la fase di *Data Mining*, in cui vengono selezioni gli algoritmi necessari per l'estrazione di pattern significativi, che possono essere suddivisi in:

- **Predizione**: ovvero la predizione di nuove variabili di interesse a partire da variabili o attributi già conosciuti.
- **Descrizione**: permette di scoprire nuovi pattern significativi per descrivere i dati.

Gli obiettivi di predizione e descrizione possono essere raggiunti usando diversi metodi:

- **Classificazione**: è utile per trainare un modello che mappa un elemento, in base alle sue caratteristiche, in una dellinsieme di classi predefinite.
- **Regressione**: è l'apprendimento di un modello predittivo che mappa un elemento in una variabile a valori reali.
- **Clustering**: utilizzato per raggruppare oggetti simili tra di loro.
- **Summarization**: racchiude tecniche che individuano una descrizione addensata dei dati analizzati.
- **Dependency Modelling**: consente di trovare un modello che descrive dipendenze significative tra variabili.
- **Change detection and deviation**: permette di scoprire variazioni preponderanti nei dati rispetto a misurazioni fatte precedentemente.

Viene di seguito descritta la tecnica di *Clustering*.

2.2.1 Clustering

Gli algoritmi di *Clustering* permettono di suddividere elementi omogenei in un insieme di dati. Fanno parte delle tecniche *unsupervised*, ovvero quelle tecniche che non necessitano di conoscenza pregressa, ma che cercano di estrarre dai dati delle correlazioni interessanti. Questi algoritmi hanno come obiettivo quello di minimizzare la distanza intracluster e di massimizzare la distanza intercluster, come mostrato in Figura 2.5.

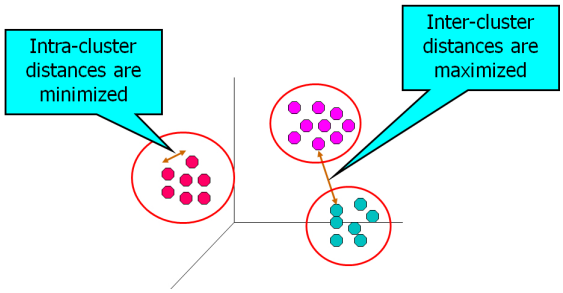

Figura 2.5: Esempio di cluster. © Tan,Steinbach,Kumar

Distanza nel Clustering

Il concetto di similarità tra un elemento e un altro è molto importante perchè permette di attribuire l'appartenenza a un *cluster* o meno. Qui di seguito viene elencata una serie di tecniche utilizzate per calcolare la distanz:

- **Distanza Euclidea**: la distanza euclidea tra due oggetti p e q è definita come la lunghezza del segmento che li congiunge:

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_\mathrm{i} - p_\mathrm{i})^2}$$

- **Distanza di Manhattan**: definisce la distanza tra due punti come la somma del valore assoluto delle differenze delle loro coordinate:

$$d(p,q) = \sum_{i=1}^{n}(|q_\mathrm{i} - p_\mathrm{i}|)$$

- **Distanza di Minkowski**: può essere considerata una generalizzazione sia della distanza Euclidea che della distanza di Manhattan. Dato un numero reale h ≥ 1, tale distanza si calcola come:

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(|q_\mathrm{i} - p_\mathrm{i})^h|}$$

Categorie di Clustering

I più diffusi algoritmi di *Clustering* possono essere categorizzati nel seguente modo [17]:

- **Gerarchico e Partizionale**: il *Clustering partizionale* è una divisione del set di dati in sottoinsiemi non sovrapposti (*cluster*), così che ogni dato appartiene ad un unico sottoinsieme.

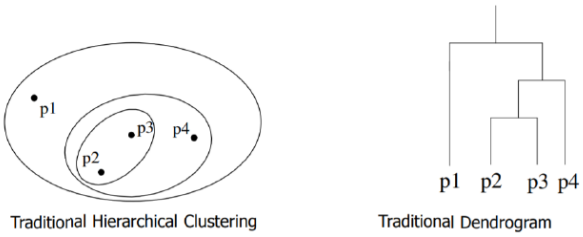

Figura 2.6: Esempio di Clustering partizionale ©Tan,Steinbach, Kumar, Introduction to Data Mining, McGraw Hill 2006

Il *Clustering gerarchico* è un set di *cluster* annidati che sono organizzati a formare un albero. La radice dell'albero è il *cluster* che contiene tutti gli oggetti, mentre ogni nodo è l'unione dei suoi figli (sottoclusters).

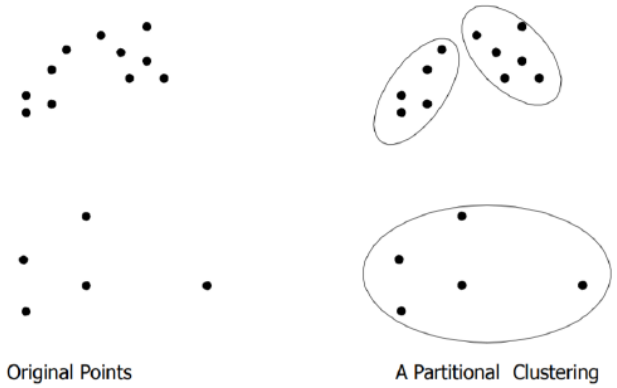

Figura 2.7: Esempio di Clustering gerarchico ©Tan,Steinbach, Kumar, Introduction to Data Mining, McGraw Hill 2006

- **Esclusivo, Non-esclusivo e Fuzzy**: un algoritmo di *Clustering* si dice *esclusivo* quando assegna ogni oggetto ad un singolo *cluster*. Invece si parla di *Clustering Non-esclusivo* quando un oggetto può appartenere simultaneamente a più di un gruppo. In un *Clustering Fuzzy* ogni oggetto può appartenere ad ogni *cluster* con un peso di appartenenza compreso tra 0 (non appartenenza) e 1 (appartenenza).
- **Completo e Parziale**: in un *Clustering completo* ogni oggetto viene assegnato ad un *cluster*, mentre in un *Clustering parziale* questo non avviene, perchè alcuni oggetti possono non appartenere a gruppi definiti.

- **Density-based**: un *cluster* è una regione densa di oggetti che è circondata da una regione a bassa densità. Una definizione basata sulla densità di *cluster* viene di solito utilizzata quando i *cluster* sono irregolari e quando sono presenti *outlier* [17].

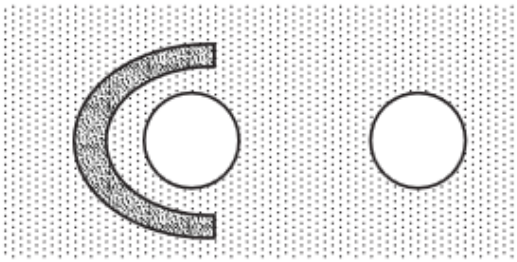

Figura 2.8: Esempio di Density-based Clustering ©Tan,Steinbach, Kumar, Introduction to Data Mining, McGraw Hill 2006

Algoritmo K-Means

Questo algoritmo fa parte della categoria di algoritmi di *Clustering* partizionali, viene utilizzato all'interno del *framework* con l'obiettivo di consentire di individuare gruppi di certificati energetici con caratteristiche simili. Il K-Means definisce per ogni *cluster*, il concetto di centroide, che corrisponde alla media di tutti i punti del *cluster*. Il primo passo del K-Means consiste nel definire il parametro K, ovvero il numero di punti scelti come centroidi iniziali.
Ogni punto è assegnato al centroide più vicino a lui e tutti i punti che sono assegnati allo stesso centroide fanno parte dello stesso *cluster*. Il centroide di ogni *cluster* è aggiornato in base ai punti assegnati al *cluster*. Si ripete l'algoritmo fintanto che i centroidi non cambiano più.
Di seguito vengono descritti i passi che compongono l'algoritmo K-Means:

Algorithm 1 K-Means

1: Seleziona K punti come centroidi iniziali
2: **repeat**
3: Forma K clusters assegnando ogni punto al suo centroide più vicino
4: Ricalcola il centroide di ogni cluster
5: **until** Centroidi non cambiano

L'insieme dei punti iniziali è scelto casualmente, per questo motivo si generano *cluster* che variano ad ogni esecuzione dell'algoritmo. Nelle Figure 2.9 e 2.10 si può vedere come, eseguendo l'algoritmo più volte sugli stessi dati, scegliendo centroidi diversi, i risultati ottenuti cambiano.

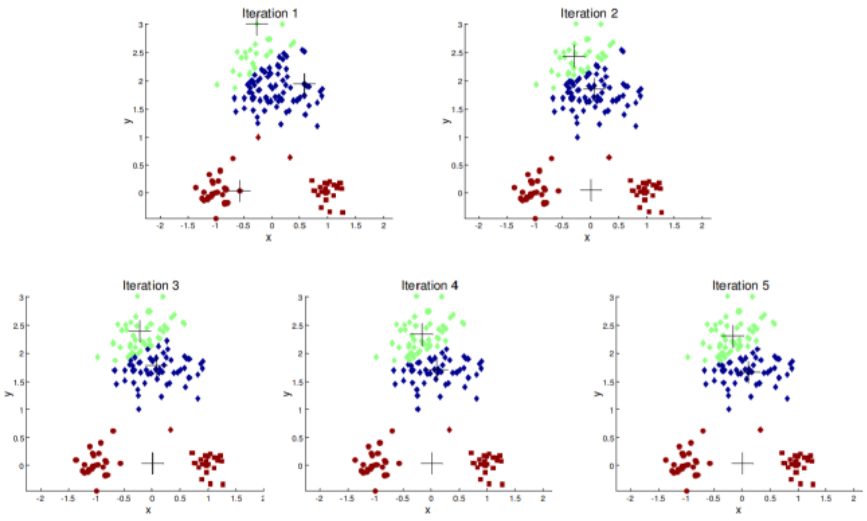

Figura 2.9: Esempio 1: Conseguenza della scelta dei centroidi © Tan,Steinbach,Kumar

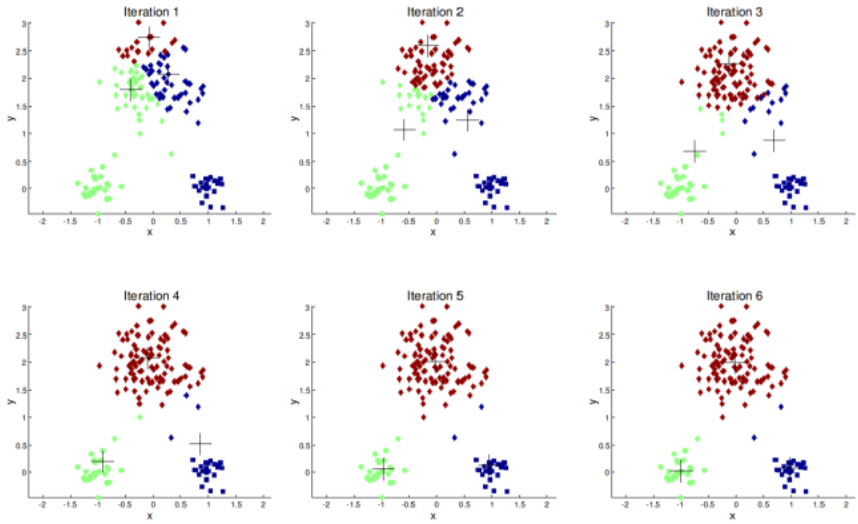

Figura 2.10: Esempio 2: Conseguenza della scelta dei centroidi © Tan,Steinbach,Kumar

La distanza Euclidea permette di misurare la differenza tra un punto p, appartenente ad un *cluster* C e il suo centroide, ed è indicata come *dist(p, ci)*. Per misurare la qualità di un *cluster* si usa la **somma dei quadrati degli errori SSE**, definita come:

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist(ci, x)^2$$

Quindi si calcola, per ogni punto, la distanza Euclidea dal suo centroide più vicino e successivamente la somma totale dei quadrati. Dati due set di *cluster* ottenuti da due diverse esecuzioni dell'algoritmo, si preferisce quello che presenta SSE più piccolo.
Lo spazio richiesto dal K-Means è ridotto, perchè occorre memorizzare solo i punti e i *cluster*. In particolare è pari a *O((m+K)n)*, dove m è il numero di punti e n il numero di attributi. Anche il tempo di esecuzione dell'algoritmo è modesto, pari a *O(I*K*m*n)*, dove *I* è il numero di iterazioni richieste per la convergenza, valore spesso basso. Il K-Means è un algoritmo semplice ed efficiente, a condizione che K sia significativamente inferiore al numero di punti. Un valido modo per ridurre l'SSE è quello di aumentare il numero di *cluster*, quindi aumentare il valore K. Tuttavia, si

preferisce non aumentare il numero di *cluster* per diminuire l'SSE, e questo è fattibile perchè il K-Means converge ad un minimo locale. Ci si focalizza sui singoli *cluster*, perchè l'SSE non è altro che la somma dell'SSE dei singoli *cluster*. Un approccio utilizzato è quello di alternare le fasi di suddivisione con quelle di fusione dei *cluster*. Due strategie che riducono l'SSE totale aumentando il numero di *cluster* sono [17]:

- **Dividere un cluster**: si sceglie di solito il *cluster* con il più grande SSE da dividere.
- **Introdurre un nuovo centroide del cluster**: spesso si sceglie il punto che è più lontano da ogni centro di *cluster*. Questo si può determinare facilmente tenendo traccia dell'SSE fornito da ogni punto.

Due strategie che invece riducono il numero di *cluster*, cercando di minimizzare l'aumento dell'SSE totale, sono le seguenti:

- **Disperdere un cluster**: operazione ottenuta rimuovendo il centroide corrispondente al *cluster* e riassegnare i punti ad altri *cluster*. Il *cluster* disperso dovrebbe essere quello che incrementa di meno l'SSE totale.
- **Unire due cluster**: si uniscono i *cluster* che hanno i centroidi più vicini, oppure quelli che risultano nel più piccolo aumento dell'SSE totale.

Determinazione del valore K

Una tecnica molto popolare per determinare il numero di *cluster* ottimale, è quella denominata *Elbow Method* [18].
Viene mostrato graficamente, al variare del numero di *cluster*, l'andamento del valore di SSE. Si vuole determinare qual è il punto in cui un aumento di K provocherà una diminuzione molto piccola del valore di SSE, mentre una diminuzione di K incrementerà bruscamente tale valore. Quindi nel punto in cui la curva ha un gomito, si trova il K ottimale.

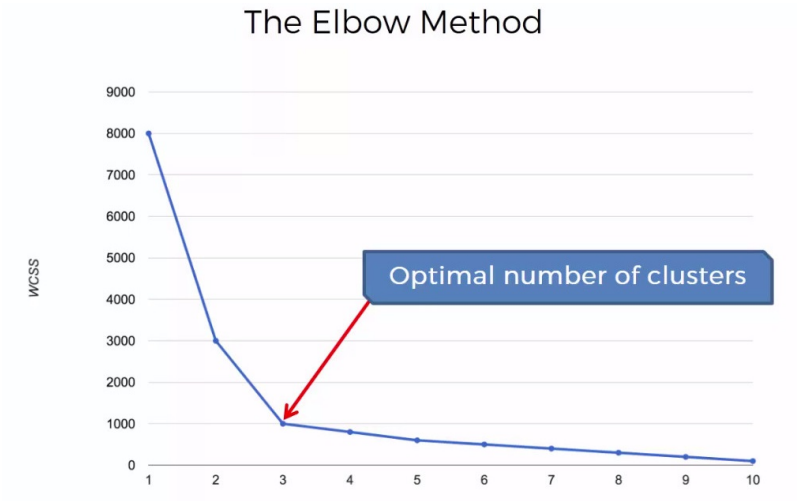

Figura 2.11: Esempio di utilizzo dell'Elbow Method

Per cercare di rendere automatica la scelta del K migliore fra quelli presenti nel gomito, si utilizza la tecnica visualizzata nella seguente figura:

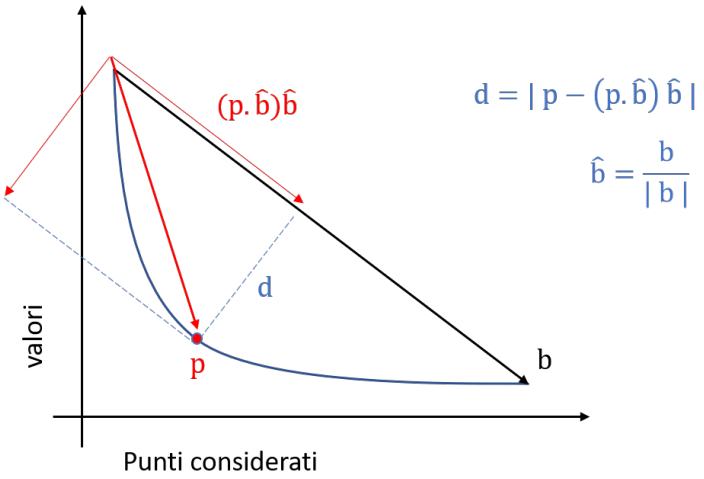

Figura 2.12: Formula per il calcolo automatico del K

Vengono calcolate le proiezioni ortogonali dei punti che appartengono alla curva sulla retta che congiunge il punto iniziale a quello finale, e si sceglie il parametro K

nell'asse delle ascisse corrispondente al punto p che ha distanza massima dalla retta che congiunge il punto iniziale e finale della curva[19].

2.3 Interpretazione della conoscenza estratta

Questa sezione viene suddivisa in due parti, la prima racchiude la *Knowledge Characterization*, che si pone come obiettivo il caratterizzare la conoscenza estratta dal blocco precedente attraverso alcuni strumenti, e la *Knowledge Visualization*, che consente di rappresentare su mappe interattive e geolocalizzate i risultati ottenuti. La seconda parte presenta l'applicazione web sviluppata, che permette di visualizzare statistiche sulle prestazioni energetiche degli edifici della città di Torino e le mappe interattive e geolocalizzate.

2.3.1 Knowledge Characterization

Il *framework* utilizza gli alberi di decisione per caratterizzare i *cluster* ottenuti, i boxplot, illustrati nella sezione 2.1, per analizzare la distribuzione delle variabili all'interno del *cluster*, e i *radar chart* per rappresentare i centroidi rispetto alle variabili di interesse ottenute dalla fase di Data seleciton.
Di seguito vengono introdotti gli alberi di decisione e i *radar chart*.

Alberi di decisione

Un albero di decisione è uno strumento che sfrutta un modello ad albero, simile ad un diagramma di flusso, in cui ogni nodo interno rappresenta un test su un attributo, ogni ramo dell'albero rappresenta un esito del test e ciascun nodo foglia rappresenta l'etichetta di classe. I percorsi dalle radici alle foglie danno luogo alle regole di classificazione. Il *CART (Classification and Regression Tree)* è un termine introdotto da Leo Breiman [20] per riferirsi a quegli algoritmi per alberi di decisione che possono essere utilizzati per problemi di modellazione predittiva di classificazione o regressione. La creazione di un albero CART implica:

- **Decidere le variabili di input**
- **Quali condizioni usare per la divisione**
- **Definire quando fermarsi**

Per scegliere le caratteristiche si utilizza di solito una tecnica che prende il nome di *Recursive Binary Splitting*. Vengono prese in considerazione tutte le caratteristiche e diversi punti di divisione vengono provati usando una funzione di costo. Viene scelta la divisione con costo più basso. Durante la prima divisione vengono considerate tutte le caratteristiche e i dati di allenamento vengono suddivisi in gruppi in base a questa suddivisione. Il CART è un algoritmo *Greedy* ed è di natura ricorsiva perchè i gruppi formati possono essere suddivisi usando la stessa strategia anche per i nodi successivi al primo.
Per stabilire un criterio di suddivisione possono essere considerate diverse euristiche, come il *Gain Ratio*, l' *Information Gain* o il *Gini Index*. Il CART fa uso del *Gini Index*, che fornisce una misura della frequenza con cui un elemento scelto casualmente dall'insieme sarebbe etichettato in modo errato se fosse etichettato in modo casuale in base alla distribuzione delle etichette nel sottoinsieme. Per calcolare il *Gini Index* di un set di oggetti con classi J, supponendo che p_i sia la frazione di oggetti etichettati con classe i nel set:

$$I_{\mathrm{G}}(p) = 1 - \sum_{i=1}^{J} p_{\mathrm{i}}^2$$

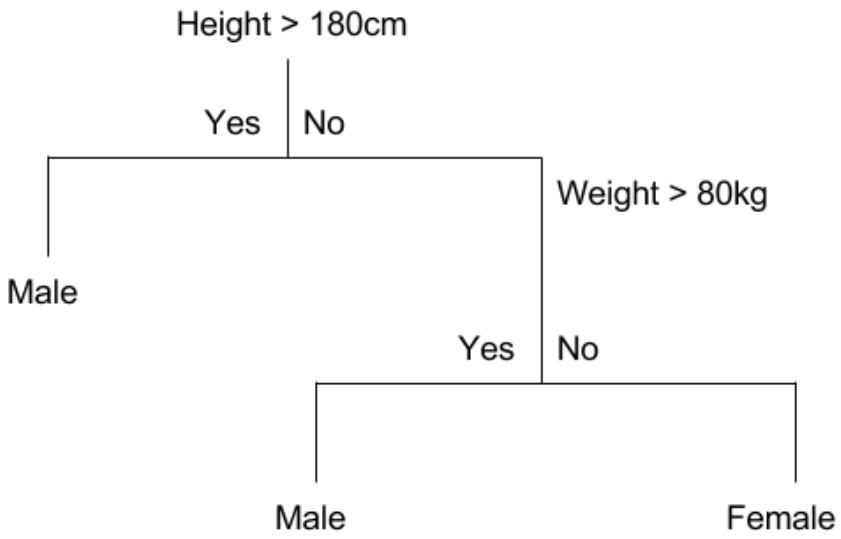

Figura 2.13: Esempio di CART

Alberi decisionali con elevato numero di nodi e di divisioni possono portare ad un *Overfitting*, cioè il modello risulta di difficile interpretazione in quanto diventa inaccurato per previsioni successive. Una tecnica per evitare questo problema è settare un numero minimo di dati di allenamento da usare per ciascun nodo foglia. Un'altra soluzione è quella di impostare la profondità massima del modello, che corrisponde alla lunghezza del percorso più lungo dal nodo radice al nodo foglia. Il *Pruning* permette inoltre di migliorare ulteriormente le prestazioni dell'albero rimuovendo i rami che fanno uso di caratteristiche con poca importanza.
Il CART può gestire dati numerici altamente distorti o multimodali, oltre a predittori categoriali, con struttura ordinale o non ordinale. Questa caratteristica permette di eliminare il tempo speso dall'analista a determinare se le variabili sono distribuite normalmente e a effettuare la trasformazione, se non lo sono. Un altro vantaggio del CART è che si tratta di un metodo di *Machine Learning* relativamente automatico, ovvero rispetto alla complessità dell'analisi, l'analista richiede un input relativamente piccolo, in netto contrasto con altri metodi in cui sono richiesti ampi input dall'analista, l'analisi dei risultati intermedi e la successiva modifica del metodo.

Classificazione

L'obiettivo della classificazione consiste nel prevedere una certa etichetta di classe, si tratta di un processo che avviene in due fasi:

- **Training**: questa fase è di tipo supervised, perchè si conosce già l'etichetta di classe. Infatti si costruisce il classificatore a partire da un insieme di righe del *dataset* che hanno già un'etichetta categorica.
- **Classification**: il classificatore realizzato nella fase precedente si occupa di predirre l'etichetta di classe di righe del *dataset* a cui non è ancora associata una *label*.

Affidabilità del modello

Un aspetto importante prima di utilizzare il classificatore consiste nel verificarne la sua affidabilità. Per questo si divide il *dataset* in due porzioni di dati, *training set* e *test set*, il primo ha l'obiettivo di costruire il modello, mentre il secondo di validarlo, costituito da tuple a cui è associata un'etichetta di classe non utilizzate per costruire il classificatore.
Prendendo in considerazione un classificatore binario, costruito per discriminare tra due classi, indicate come positiva e negativa, a partire da un insieme di oggetti di cui si conosce già l'etichetta di classe (P = positiva e N = negativa), si può valutare l'affidabilità del classificatore calcolando il numero di casi classificati in modo errato. Vengono definiti i seguenti termini:

- **True Positive (TP)**: rappresentano tutti gli oggetti classificati come P e lo sono realmente.
- **True Negative (TN)**: rappresentano tutti gli oggetti classificati come N e lo sono realmente.
- **False Positive (FP)**: rappresentano tutti gli oggetti classificati come P e non lo sono realmente.
- **False Negative (FN)**: rappresentano tutti gli oggetti classificati come N e non lo sono realmente.

Questi dati vengono posti nella cosiddetta matrice di confusione, che rappresenta l'accuratezza di un modello di classificazione:

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Figura 2.14: Esempio di matrice di confusione

Si definiscono alcune misure di performance: l'accuratezza, che misura la percentuale di tuple del *dataset* che sono state correttamente classificate dal classificatore, definita come:

$$Accuratezza = \frac{TN + TP}{N + P}$$

La precisione, che misura la percentuale di tuple che sono state classificate come P e che lo erano veramente:

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

Il richiamo, che misura la percentuale di tuple che sono state etichettate come P ed assegnate in questo modo dal classificatore.

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

Alcune tecniche di partizionamento dei dati vengono qui proposte:

- **Holdout**: i dati vengono suddivisi in modo random in due insiemi indipendenti, *training set* e *test set*. Tipicamente il *test set* ha dimensione inferiore

del *training set*.

- **Cross-Validation (k-fold)**: il *dataset* viene suddiviso in modo random in k sottoinsiemi di uguale dimensione. Dei k sottoinsiemi, uno viene utilizzato come validation set e i rimanenti k-1 come training. Viene utilizzato per eliminare il problema dell'overfitting nei *training set*.

- **Leave-one-out cross validation**: la *leave-one-out cross validation* utilizza un'unica osservazione del *dataset* originario come dato di *test* e le restanti osservazioni come dati di *training*. È equivalente al *k-fold cross-validation* con k uguale al numero di dati nel *dataset* di partenza.

Boxplot

Si tratta di un metodo grafico per rappresentare una distribuzione statistica di un campione mediante indici di dispersione e di posizione. Viene rappresentato mediante un rettangolo, che può essere visualizzato orizzontalmente o verticalmente, da cui fuoriescono due segmenti, rappresentanti il massimo e il minimo. La linea interna al rettangolo prende il nome di *mediana*, le linee estreme vengono chiamate primo e terzo quartile. La distanza tra il primo ed il terzo quartile, chiamata *distanza interquartilica*, misura la dispersione della distribuzione e contiene il 50% delle osservazioni. I valori esterni al valore massimo e minimo rappresentano gli *outlier* [21].

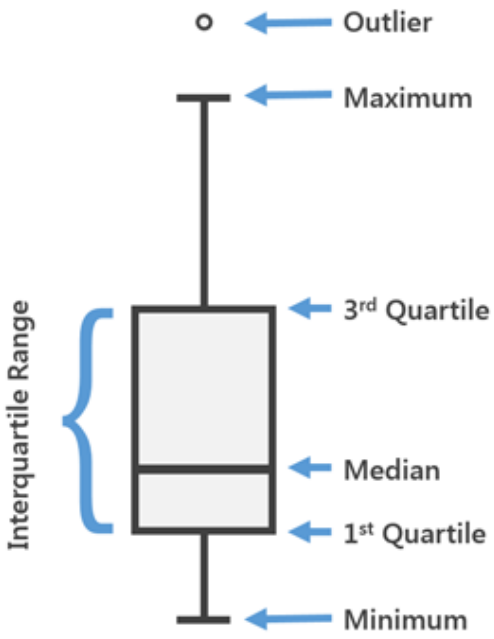

Figura 2.15: Esempio di un Boxplot

Radar chart

Un *radar chart* è un metodo grafico per visualizzare dati su variabili multiple in forma di grafico a 2-dimensioni, di 3 o più variabili, rappresentate su assi con l'origine in comune. Rappresenta un modo per comparare tra di loro più variabili, per vedere quali variabili hanno valore simile e se sono presenti *outlier* tra ciascuna variabile. È costituito da raggi equi-angolari, che rappresentano una variabile di interesse. Ogni valore di variabile viene tracciato lungo il proprio asse individuale e tutti i valori delle variabili sono collegati tra di loro da segmenti a formare un poligono. Qui di seguito viene mostrato un esempio di *radar chart*:

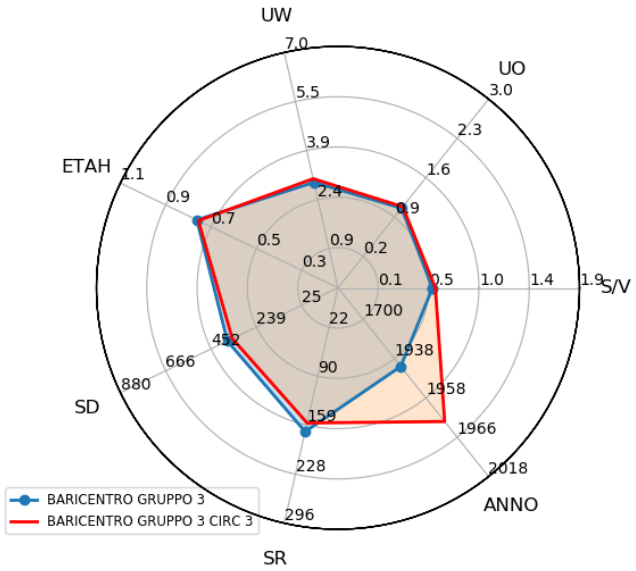

Figura 2.16: Esempio di *radar chart*

Alcuni svantaggi dei *radar chart* riguardano il numero di variabili da visualizzare, che se è in un numero troppo elevato, rende il grafico difficile da leggere, quindi è buona pratica mantenere limitato il numero di variabili. Un altro svantaggio riguarda la presenza di poligoni multipli, la cui lettura diventa difficoltosa se i poligono si sovrappongono tra di loro.

2.3.2 Knowledge Visualization

La visualizzazione della conoscenza estratta avviene attraverso tre tipologie di mappe geolocalizzate interattive: mappe *coropletiche*, mappe *scatter* e mappe *marker-cluster*. Questa tipologia di visualizzazione rende fruibile la conoscenza estratta anche ai non esperti di dominio e permette di avere una visione a differente granularità spaziale: circoscrizione, quartiere e singolo edificio. Di seguito viene data una spiegazione per ogni tipologia di mappa:

Mappe coropletiche

Le mappe coropletiche sono mappe tematiche, in cui le aree delimitate all'interno di esse, sono colorate in proporzione della misurazione della variabile che si intende visualizzare. Forniscono un modo semplice per capire come varia una misura

attraverso un'area geografica. La variabile utilizza la progressione del colore per rappresentarsi su ciascuna sezione della mappa. Può essere una fusione da un colore ad un altro, una progressione di tonalità singola, trasparente a opaco, da chiaro a scuro o un intero spettro di colori. Hanno la capacità di rappresentare una grande quantità di dati su qualsiasi quantità di spazio in modo succinto e gradevole alla vista. Tuttavia, non è il metodo ideale per rappresentare realisticamente i dati, perchè questa tipologia di mappa ha il limite di non riuscire a mostrare bene la vera fluttuazione delle statistiche in tutta l'area. Di seguito un esempio di mappa coropletica:

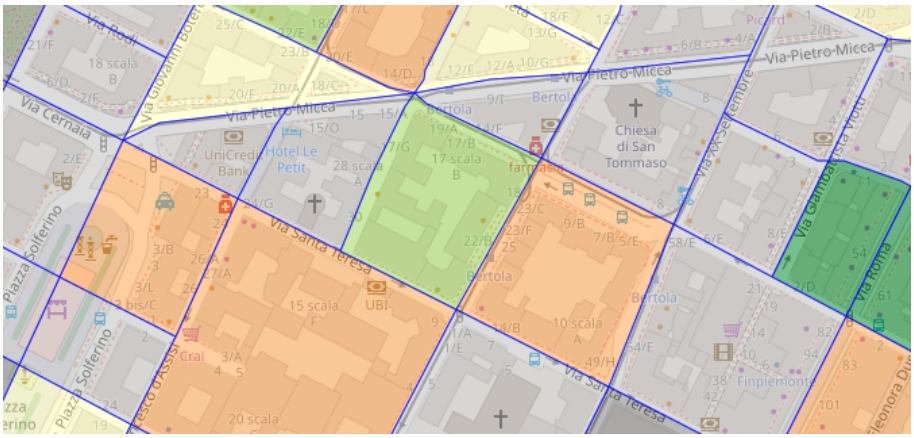

Figura 2.17: Esempio di mappa coropletica

Mappe scatter

Le mappe di tipo *scatter* permettono di visualizzare dati geografici come marker (punti) sulla mappa. Questo tipo di informazione visualizzata è molto dettagliato ma presenta lo svantaggio di rendere poco agevole la navigazione nel caso in cui il numero di marker visualizzati sia molto elevato. Di seguito un esempio di mappa *scatter*:

Figura 2.18: Esempio di mappa *scatter*

Mappe marker-cluster

Questa tipologia di mappa rende visualizzabili i marker in maniera aggregata, attraverso dei cerchi, la cui dimensione riflette il numero di marker contenuti all'interno e il cui colore cambia in base al valore medio dei punti considerati per l'aggregazione. Le mappe coropletiche sono molto utili per visualizzare la distribuzione geografica di una sola variabile, le mappe *marker-cluster* invece consentono di rappresentare più variabili allo stesso tempo. I marker sono visualizzati solo quando si raggiunge un certo livello di zoom sulla mappa e sono raggruppati nuovamente in *cluster* quando si esegue uno zoom all'indietro. Questa tipologia di mappa presenta il vantaggio di poter mettere un elevato numero di marker su una mappa evitando di renderla difficile da navigare. Di seguito un esempio di mappa *marker-cluster*:

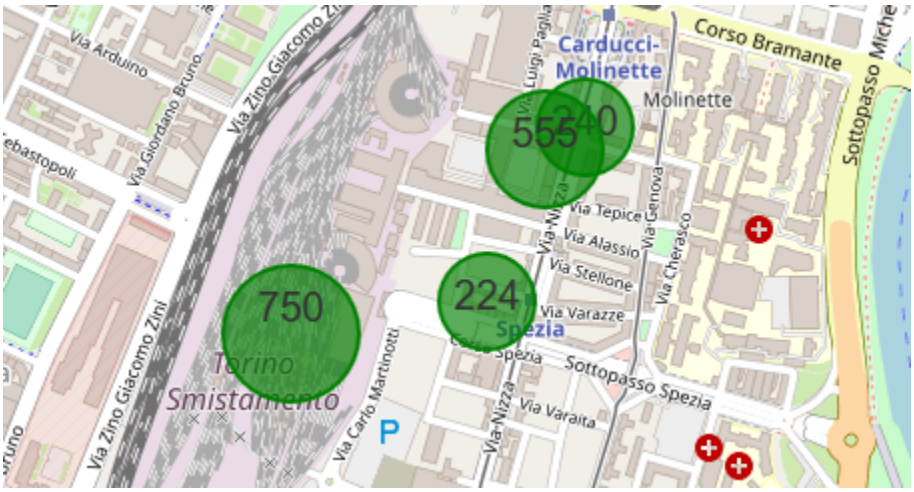

Figura 2.19: Esempio di mappa *marker-cluster*

2.3.3 Applicazione web

L'applicazione web sviluppata ha come obiettivo il visualizzare in maniera dinamica le statistiche sulle prestazioni energetiche degli edifici della città di Torino e le mappe interattive e geolocalizzate illustrate nella sezione precedente.
Un applicazione web sfrutta un'architettura *client-server*, dove per *client* si intende un programma che effettua richieste di servizi ad un altro programma, chiamato *server*. Il client accede tramite un *web browser* a funzionalità applicative giacenti su un *application server*. Il codice che viene eseguito sulla parte client costituisce il *front-end* dell'applicazione web, eseguito dal *web browser* e caratterizzante l'interfaccia utente. La parte di *front-end* è stata realizzata utilizzando il linguaggio *HTML5*, *Javascript* e *CSS3* [22]:

- **HTML (HypterText Markup Language)**: è un linguaggio di *markup* per la formattazione e paginazione di pagine web. La sua sintassi è definita dal *World Wide Web Consortium (W3C)*. Non è un linguaggio di programmazione perchè non prevede alcuna definizione di variabili, strutture dati, funzioni o strutture di controllo, ma si occupa di gestire i contenuti associandone il layout all'interno della pagina web grazie all'utilizzo di *tag*.
- **Javascript**: è un linguaggio di *scripting* orientato agli oggetti e agli eventi che permette di realizzare effetti dinamici interattivi tramite funzioni di script, opportunamente inserite nei file HTML o in appositi file separati, richiamate al momento dell'utilizzo all'interno delle pagine web.

- **CSS (Cascading Style Sheets)**: è un linguaggio utilizzato per definire lo stile (font, colori, spazi) di pagine web. È fatto da regole, ognuna delle quali si applica ad un particolare elemento HTML e controlla un determinato aspetto del suo rendering. Il CSS è stato introdotto per separare i contenuti delle pagine HTML dal loro layout e permettere una programmazione più chiara.

Il codice che viene eseguito lato *server*, prende il nome di *back-end*, che riceve richieste dai *client* e contiene la logica per inviare i dati in risposta. Il *back-end* include anche il *database* per memorizzare in modo persistente tutti i dati dell'applicazione web. La parte di *back-end* dell'applicazione web è stata sviluppata utilizzando *Flask* [23]. **Flask** è un micro *web-framework* scritto in Python, micro perchè ha un nucleo semplice ma espandibile con estensioni a piacere. Ha due dipendenze principali: il *routing*, il *debugging* e il *Web Server Gateway Interface (WSGI)* provengono da Werkzueg, mentre il motore di *template* è fornito da Jinja2. I *template* sono file scritti in linguaggio non puro HTML, che contengono sia dati statici che *placeholder* e variabili per i dati dinamici. Quando vengono renderizzati da un motore di template, i contenuti dinamici vengono sostituiti dai loro valori attuali, producendo un file HTML definitivo.
I *client* inviano richieste al web *server*, la quale li invia all'istanza di applicazione Flask. L'istanza dell'applicazione ha bisogno di sapere quale codice eseguire per ogni URL richiesto, quindi mantiene una mappatura interna tra gli URL e le funzioni Python. Una *route* è un'associazione tra un'URL e la funzione che la gestisce. Il modo più conveniente per definire una route in un'applicazione Flask avviene attraverso l'*app.route.decorator*. L'utilizzo di Flask presenta come vantaggi:

- Design leggero e modulare: facile da trasformare nel *web-framework* desiderato con poche estensioni senza appesantirlo.
- *ORM-agnostic*: si può collegare l'ORM preferito.
- Documentazione completa, ricca di esempi e ben strutturata.
- Elevata flessibilità di configurazione.
- Semplice da imparare ed utilizzare.

2.4 Generalizzazione della conoscenza

Lo scopo ultimo non è solo caratterizzare la conoscenza ma anche la sua generalizzazione a nuovi dati. Nel *framework* sono stati sviluppati due modelli di generalizzazione: semi-completo e completo.

2.4.1 Modello completo e semi-completo

Qualora i nuovi dati in ingresso presentino valori mancanti tra i loro attributi, si applica il modello completo, predicendo tali valori attraverso tecniche di regressione (o statistica referenziale), per poi successivamente predirre l'etichetta di *Clustering*. Se invece i nuovi dati si presentano completi, si segue esclusivamente il modello semi-completo, che si occupa della loro classificazione. Di seguito viene fornita una panoramica su alcuni algoritmi di regressione e sul principale metodo utilizzato per la classificazione.

Tecniche di regressione

Di seguito vengono descritti alcuni tra i metodi di regressione principali:

- **Lineare**: la regressione lineare [24] è un metodo di stima del valore atteso di una variabile dipendente Y, dati i valori di altre variabili indipendenti $X_1, ..., X_\mathrm{k}$. Se k è uguale a 1, si parla di regressione lineare semplice, altrimenti di regressione multipla. Il modello più generale assume la seguente forma:

  $$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_\mathrm{k} + \epsilon$$

  Dove X_i denota l'i-esimo predittore, mentre β_i quantifica l'associazione tra X_i e Y. I parametri $\beta_0, ..., \beta_k$ sono incogniti e vanno stimati secondo la seguente formula:

  $$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + ... + \hat{\beta}_k x_\mathrm{k}$$

  Dove $\hat{y}$ indica una previsione di Y, dipendente dai valori $X_1 = x_1, ..., X_\mathrm{k} = x_\mathrm{k}$. Per dedurre i parametri della funzione lineare esistono vari metodi, tra cui il metodo dei minimi quadrati, che minimizza, selezionando i coefficienti

$\beta_0, ..., \beta_k$, la somma dei residui quadratici RSS:

$$RSS = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

- **Polinomiale**: il legame tra la variabile Y e gli attributi $X_1, ..., X_k$ potrebbe non essere lineare. Pertanto la regressione polinomiale estende quella lineare utilizzando l'interpolazione polinomiale [25] dei dati. Il modello di regressione polinomiale assume la seguente forma:

$$Y = \beta_0 + \sum_{i=1}^{k} f_i(X_i)$$

  La relazione non lineare tra ogni *feature* e la variabile di risposta si ottiene andando a sostituire ogni componente lineare $\beta_i X_i$ con una funzione polinomiale $f_i(X_i)$ di grado d.

- **Ridge**: la regolarizzazione di Ridge [26] è un metodo molto simile ai *minimi quadrati lineari*, che consentono di trovare i Beta stimati minimizzando la somma dei quadrati dei residui (RSS). Per stimare i Beta, si somma all'RSS il termine di *penalità*, pari alla sommatoria dei coefficienti Beta al quadrato, moltiplicata per il termine di *tuning* $\lambda(\geq 0)$:

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = RSS + \lambda \sum_{j=1}^{p} \beta_j^2$$

  Per non avere penalità bisogna che si presenti $\lambda = 0$, dall'altra parte $\lambda \to +\infty$ porterà ad avere una penalità molto elevata, ovvero molti coefficienti saranno prossimi a zero. Aumentando λ il modello sarà meno flessibile, con la conseguenza di un bias maggiore e una varianza minore. Se $p < n$, ovvero il numero di predittori è elevato ma minore della numerosità, il metodo dei minimi quadrati può trovare difficoltà a causa dell'elevata variabilità. Se invece $p > n$, cioè il numero di predittori è superiore alla numerosità del campione, il metodo dei minimi quadrati può essere utilizzato. Il metodo Ridge invece può essere usato in entrambi i casi ma presenta lo svantaggio che non permette mai l'esclusione dei Beta simili a 0 dal modello.

- **Lasso**: Nella statistica e nel *Machine Learning*, il metodo Lasso (*Least Absolute Shrinkage and Selection Operator*) è un metodo di analisi di regressione che esegue sia la selezione della variabile che la regolarizzazione, al fine di migliorare l'interpretabilità e l'accuratezza della previsione del modello statistico che produce. È stato originariamente introdotto nella letteratura geofisica nel 1986 [27] e successivamente reso popolare da Robert Tibshirani nel 1996 [28]. Lasso è stato formulato inizialmente per i modelli dei minimi quadrati ed esteso ad un'ampia varietà di modelli statistici, tra cui modelli lineari generalizzati, equazioni di stima generalizzate e modelli di rischi proporzionali. Rispetto al metodo Ridge, permette ai coefficienti dei beta stimati di essere esclusi dal modello quando sono pari a zero. La formula di Lasso è molto simile a quella di Ridge, a differenza che la sommatoria è calcolata sul valore assoluto dei beta:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = RSS + \lambda\sum_{j=1}^{p}|\beta_j|$$

  Grazie alla presenza del valore assoluto, alcuni coefficienti vengono posti esattamente pari a zero, questo si presenta quando λ è molto elevato. Come nel caso di Ridge, per non avere penalità e quindi avere una stima analoga a quella predetta dai minimi quadrati, bisogna che λ sia prossimo a zero. Quando $\lambda \to +\infty$, tutti i coefficienti sono pari a zero, ad esclusione dell'intercetta, che è stata esclusa a priori da questa tecnica. Sia per la regressione Ridge che per Lasso, due buoni metodi per determinare il parametro di tuning λ è la *cross validation* e la *grid search*.
- **K-NN**: le regressioni polinomiali e lineari hanno lo svantaggio di presupporre la forma della funzione $f(X_1, ..., X_k)$. Se l'ipotesi sulla forma non corrisponde alla realtà, la predizione della variabile *target* sarà poco precisa. Il metodo K-NN (K-Nearest Neighbors) [29] non necessita a priori di informazioni sulla forma della funzione incognita f. Essa identifica per ogni nuova osservazione x_0 le k osservazioni del *dataset* più prossime ad essa (cioè il vicinato N_0), per poi procedere con la stima di $f(x_0)$ data la media dei valori della variabile in N_0.

Per confrontare la bontà dei metodi di regressione si può usare il *coefficiente di determinazione* R^2 [30] calcolato sul *dataset* di training. R^2 misura la frazione della

varianza della variabile dipendente espressa dalla regressione, cioè partendo dalla retta di regressione, sintetizza in un unico valore, di quanto la grandezza analizzata si discosta mediamente da tale retta:

$$R^2 = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS}$$

Dove TSS è pari a:

$$TSS = \sum_{i=1}^{n}(y_i - \overline{y})^2$$

y_i sono i dati osservati, mentre $\overline{y}$ è la loro media. R^2 può assumere valori compresi tra 0 e 1, se è pari a 1, esiste una perfetta relazione lineare tra il fenomeno analizzato e la sua retta di regressione. In generale, più alto è R^2, meglio il modello si adatta ai dati. Nella seguente figura sono mostrati due modelli di regressione, quello a sinistra rappresenta il 38% della varianza, mentre quello a destra rappresenta l'87% della varianza. Più è alta la varianza che è rappresentata dal modello di regressione, più i punti cadranno vicini alla linea di regressione.

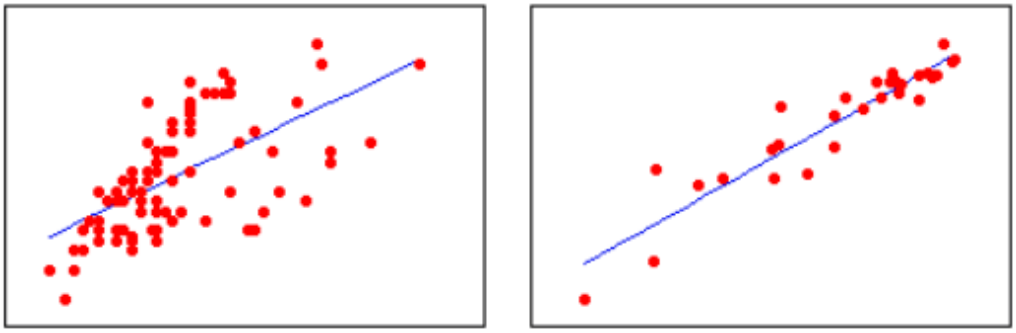

Figura 2.20: Modelli di regressione a confronto

KNN

Il metodo utilizzato dal *framework* per predirre l'etichetta di *Clustering* per nuovi dati in ingresso, sia per il modello completo che per il semi-completo, è il K-Nearest Neighbors (KNN).

Il KNN è un algoritmo adatto a risolvere sia problemi di regressione (come visto in precedenza) che di riconoscimento di pattern per la classificazione [31]. Nei problemi di classificazione, viene scelta la *label* di classe, in base ad un tipo di distanza. Tra quelle più utilizzate, oltre alla distanza Euclidea, si ha la distanza di *Manhattan* o la distanza di *Minkowsky*. Dopo aver calcolato la distanza, viene restituita l'etichetta della classe di maggioranza dell'insieme delle istanze k selezionate. Per quanto riguarda il problema della regressione, il KNN sceglie il valore medio dei valori dei vicini più prossimi come valore predetto. Il primo passo dell'algoritmo KNN è la scelta del parametro K, tra i metodi più diffusi si utilizza la *cross validation*. Il passo successivo consiste nel calcolare la distanza tra il nuovo dato e quelli del *training set*. Si ordinano le distanze ottenute dalla più piccola alla più grande e si scelgono le prime K. Se il problema è di regressione, si calcola la media dei valori K. Se invece il problema è di classificazione, si sceglie la classe di maggioranza. L'algoritmo KNN è tanto più oneroso tanto più è grande il *training set*, bisogna comunque trovare un *trade-off*, perchè un training-set ampio è tendenzialmente più rappresentativo.

Capitolo 3

Risultati sperimentali

In questo terzo capitolo vengono descritti il setting dei parametri e i risultati sperimentali ottenuti per ciascun blocco del *framework*. Prima di passare ad una descrizione dettagliata, di seguito vengono descritti gli strumenti utilizzati.

Il *framework* è stato sviluppato in Python[1], utilizzando l'ambiente di sviluppo integrato PyCharm[2], più una serie di librerie di seguito elencate.

Libreria	**Descrizione**
pandas[32]	Lettura/Modifica di Dataframe
matplotlib[33]	Generazione di istogrammi, boxplot, radar chart, ecc...
scikit-Learn[34]	*Clustering*, predizione e classificazione
folium[35]	Realizzazione di mappe geolocalizzate
shapely	analisi ed elaborazione di oggetti geometrici nel piano cartesiano
pygeoj, pyproj	scrittura, lettura e modifica di file geojson
geocoder	gestione di servizi per la risoluzione di indirizzi

Tabella 3.1: Panoramica di alcune librerie Python utilizzate dal *framework*

Il *framework* si è servito anche del linguaggio di programmazione R[36] per effettuare

[1] https://www.python.org/

[2] https://www.jetbrains.com/pycharm/

alcune statistiche. Di seguito viene mostrato un riepilogo di alcune librerie utilizzate per questo linguaggio.

Libreria	Descrizione
ggplot	generazione di boxplot
rpart	costruzione di alberi di regressione e classificazione, ad es: CART
distances	calcolo della distanza spaziale e della similarità
MLmetrics	metriche di performance dei modelli costruiti

Tabella 3.2: Panoramica di alcune librerie R utilizzate dal *framework*

Altri strumenti di supporto sono stati utili per le analisi, quali Excel[3] e Rapidminer[37], quest'ultimo è un software *open-source* che consente di applicare tecniche *Data Mining* in maniera automatica.
Per quanto riguarda l'applicazione web, come già anticipato nel paragrafo 2.3.3, la parte di *front-end* è stata realizzata in HTML5, Javascript e CSS3, mentre il back-end attraverso l'utilizzo di Flask [4].

3.1 Data Preprocessing

In questa sezione viene illustrata la fase di *Data Cleaning* e i relativi risultati ottenuti.

3.1.1 Data Cleaning

Address resolution

La pulizia degli indirizzi dei certificati energetici presenti nel *dataset* è un'operazione molto importante, perchè da questa fase deriva la correttezza dei risultati visualizzati attraverso mappe geolocalizzate interattive. Gli indirizzi del *dataset* presentavano problemi su questi aspetti: errori di battitura, caratteri codificati non correttamente, presenza del CAP generico 10100 e coordinate di latitudine e longitudine errate.

[3]https://products.office.com/it-it/excel

[4]http://flask.pocoo.org/

Inoltre per le certificazioni ACE, non erano presenti i campi relativi al CAP e alle coordinate. Per risolvere questi problemi e ottenere un *dataset* pulito, si è provveduto ad applicare l'algoritmo di risoluzione degli indirizzi basato sul calcolo dell'indice di similarità di *Levenshtein*. Qui di seguito viene riportato lo pseudocodice della parte più importante dell'algoritmo:

Algorithm 2 Algoritmo di risoluzione degli indirizzi

Input: viaUtente, dizionarioViario
Output: viaTrovata, maxLevValue

```
1:  maxLevValue ← 0
2:  lenviaUtente ← lunghezza viaUtente
3:  for viaViario, listaPermutazioni in dizionarioViario do
4:    levValue ← 0
5:    for Permutazione in listaPermutazioni do
6:      lenPermutazione ← lunghezza permutazione
7:      if levValue !=1  and lenviaUtente ≥ lenPermutazione then
8:        viaConcatenata ← Concatena parole Permutazione
9:        simValue ← Calcola simLev tra viaUtente e viaConcatenata
10:       if simValue > levValue then
11:         levValue ← simValue
12:       end if
13:     end if
14:   end for
15:   if levValue ≥ maxLevValue then
16:     maxLevValue ← levValue
17:     Aggiungi viaViario alla listaVie
18:     Aggiungi maxLevValue alla listaLevenshtein
19:   end if
20: end for
21: viaTrovata ← prendi ultima via in listaVie
22: diff ← calcola differenza tra lunghezza viaUtente e viaTrovata
23: while listaLevenshtein[indiceviaTrovata] == maxlevValue do
24:   viaTrovata ← listaVie[viaTrovata-1]
25:   if differenza tra lunghezza viaUtente e viaTrovata < diff then
26:     Aggiorna diff
27:     Aggiorna viaTrovata
28:   end if
29: end while
30: return viaTrovata, maxLevValue
```

Gli input richiesti dall'algoritmo sono l'indirizzo da risolvere (*viaUtente*) e un dizionario, che ha come chiavi gli indirizzi presenti nel viario di Torino e come valori corrispondenti le lista delle permutazioni di tali indirizzi.
Per ogni via da risolvere, si pone inizialmente a zero l'indice di similarità di Levenshtein da ottenere (*maxLevValue*). Si cicla per ogni indirizzo presente nel viario e per ognuno di esso, si determina il valore dell'indice di similarità di Levenshtein più grande (*levValue*) tra ogni sua permutazione e *viaUtente*. Per ogni via presente nel viario si controlla se *levValue* è maggiore o uguale a *maxLevValue*, se lo è si salva nella lista *listaLevenshtein* il suo valore, la via del viario nella lista *listaVie* e si aggiorna il valore di *maxLevValue* al valore di *levValue*. Non appena si finisce di scandire tutto il viario di Torino, si procede con il determinare all'interno di *listaVie* quella via con il numero di parole più vicino alla via oggetto della risoluzione e che presenta il valore massimo di Levenshtein tra quelli presenti in *listaLevenshtein*. Quest'ultimo passaggio è essenziale per evitare che un indirizzo come: "via Po 15", sia associato ad esempio a: "via Monteu da Po 15". Infatti "via Monteu da Po" ha esattamente "via Po" tra le sue permutazioni. Per questo motivo è fondamentale prendere l'indirizzo con il numero di parole più vicino tra tutti quelli memorizzati in *listaVie*. L'algoritmo alla fine restituisce la via risolta *viaTrovata* e *maxLevValue*. Se l'indice di similarità ottenuto è superiore a 0.95 (valore stabilito a seguiti di diverse prove sperimentali), l'indirizzo viene sostituito con quello ottenuto dal viario di Torino, assieme al CAP, alle coordinate e al numero civico.
Un esempio di risoluzione degli indirizzi è presentato nella Tabella 3.3. Il campo Indirizzo Input e Civico Input indicano rispettivamente il campo indirizzo e il campo numero civico presenti allinterno del *dataset*.

Indirizzo Input	**Civico Input**	**Indirizzo Output**	**Civico Output**	**Lev**
VIPACCO	45	VIA VIPACCO	45	1
CORSO ROSSELLI	155	CORSO CARLO E NELLO ROSSELLI	155	1
CORSO E. GAMBA	36-S SCALA S	CORSO ENRICO GAMBA	36	0.95
CORSO MARONCELLI PIETRO	9	CORSO PIERO MARONCELLI	9	0.98

Tabella 3.3: Esempio di risoluzione indirizzi con Levenshtein

Qualora l'indice di similarità ottenuto fosse inferiore allo 0.95, viene effettuata una richiesta al servizio di Google Geocoding. In figura 3.4 sono mostrati alcuni esempi di indirizzi risolti dall'API Google.

Indirizzo Input	Civico Input	Indirizzo Output	Civico Output	Lev
VIA DAUBR	10	VIA ADOLPHE DAUBREE	10	0.84
VIA BOVIO	3	VIA BOBBIO	3	0.88
CORSO SIRACURA	166	CORSO SIRACUSA	166	0.92
VIA MARTIGNANA	25	VIA MARTINIANA	25	0.90
C. SO SICILIA	21	CORSO SICILIA	21	0.86

Tabella 3.4: Esempio di risoluzione indirizzi con Google Geocoding

A partire da un *dataset* contenente circa 50.000 certificati, l'algoritmo basato sull'indice di similarità di Levenshtein ha potuto risolvere il 96.3% degli indirizzi. Per gli indirizzi con numero civico assente, è stato assegnato quello più frequente per le vie presenti nel viario, mentre in assenza di numero civico, è stato assegnato quello più prossimo. Il servizio di Google Geocoding ha risolto il 3,5% degli indirizzi. Complessivamente sono stati scartati lo 0,20% di certificati, ovvero quelli che avevano coordinate fuori la città di Torino e quelli che avevano un campo indirizzo vuoto.

Expert-Driven outlier detection

A seguito della pulizia degli indirizzi, si passa alla fase di *outlier detection*, guidata dall'esperto di dominio, sugli attributi selezionati e ritenuti critici per la determinazione della prestazione energetica degli edifici su cui applicare l'*outlier detection*. Gli attributi selezionati sono i seguenti:

- **Fattore forma**
- **Trasmittanza media delle superfici trasparenti**
- **Trasmittanza media delle superfici opache**
- **Rendimento di distribuzione**
- **Rendimento di regolazione**
- **Rendimento di generazione**
- **Rendimento di emissione**

È stato effettuato uno *scaling* dei valori relativi ai rendimenti, perchè molti di essi manifestavano valori corretti ma espressi su una scala errata. Nel caso in cui un rendimento assumeva un valore maggiore di 1, è stata effettuata una divisione per un fattore 100. Per gli impianti di *teleriscaldamento* un valore fisso di 0,945 è stato stabilito come *rendimento di generazione*. Tale valore è stato percepito da un

documento rilasciato da IREN [38]. Per gli impianti con pompa di calore, valori con *rendimento di generazione* maggiore ad 1 e dell'ordine della decina sono stati considerati ammissibili. Dopo queste operazioni sono stati applicati i filtri stabiliti dall'esperto di dominio.

Attributo	**Range di ammissibilità**
Fattore Forma	[0.1 - 2]
Rendimento Distribuzione	[0.75 - 1.25]
Rendimento Emissione	[0.85 - 1]
Rendimento Generazione	[0.65 - 1.1]
Rendimento Regolazione	[0.6 - 1]
Trasmittanza Opaca	[0.1 - 3]
Trasmittanza Trasparente	[0.9 - 7]

Tabella 3.5: Filtri sulla base dei range di ammissibilità

Inizialmente sono stati applicati i filtri relativi al *fattore forma* e alle *trasmittanze opache* e *trasparenti*. Questi hanno escluso circa il 2% dei certificati dal *dataset*. In Figura 3.1 è mostrato un dettaglio delle distribuzioni delle trasmittanze trasparenti e opache prima dell'applicazione dei filtri. Le linee verticali rosse indicano il range di ammissibilità.

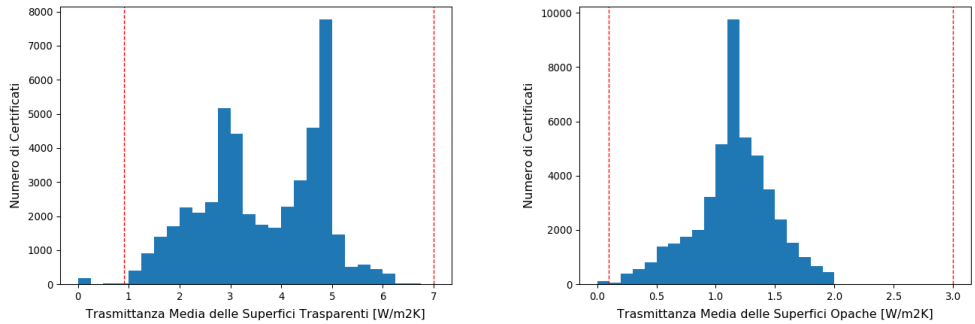

Figura 3.1: Distribuzione delle trasmittanze trasparenti e opache e filtri applicati

Per quanto riguarda l'applicazione dei filtri sui 4 rendimenti, questi hanno portato ad escludere un maggior numero di certificati. Sono rimasti a disposizione circa il

66% dei certificati originali. La Figura 3.2 mostra un dettaglio delle distribuzioni dei rendimenti prima dell'applicazione dei filtri.

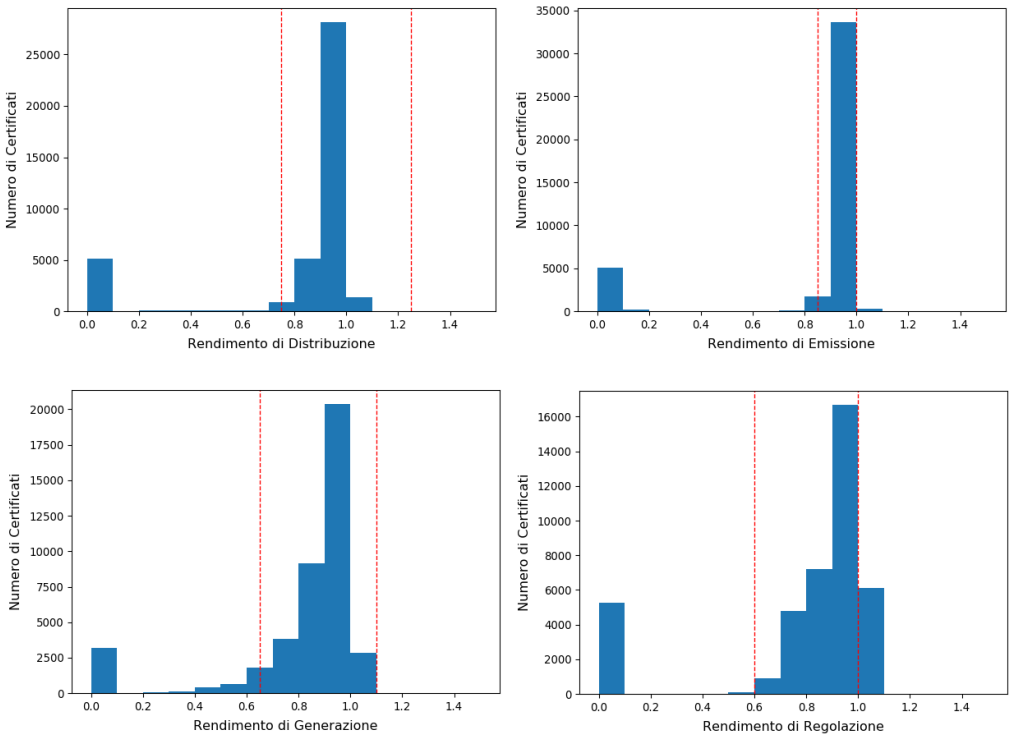

Figura 3.2: Distribuzione dei rendimenti e filtri applicati

Il numero elevato di certificati esclusi è da attribuire a valori fuori range, ma anche alla presenza di valori nulli.

Univariate outlier detection

Si sono applicate le tecniche di *outlier detection*: *gESD* e *percentile*, su tutte le altre variabili interessanti per l'analisi, per cui non sono stati definiti degli intervalli di ammissibilità.

gESD

L'applicazione del *gESD* ha riguardato degli attributi rilevanti per l'analisi, tra cui:

- **Superficie riscaldata**
- **Superficie disperdente**
- **Anno di costruzione**
- **Volume lordo riscaldato**
- **Emissioni gas serra**
- **Anno di costruzione**
- **Fabbisogno di energia termica utile**

Il parametro di *significatività (alpha)* è stato posto all'1% del numero di certificati presenti nel *dataset*, mentre il parametro *MaxOLs* è stato posto pari allo 0.5%. Questa fase di rimozione degli *outlier* ha portato il *dataset* a circa 30.000 certificati.

Percentile outlier detection

Il *gESD* ha ottenuto degli ottimi risultati sull'intervallo superiore delle distribuzioni di alcune variabili geometriche, come la superficie riscaldata, la superficie disperdente e il volume lordo riscaldato. Sull'intervallo inferiore alcuni valori non ammissibili sono rimasti, si è provveduto quindi a rimuovere quei valori al di sotto del percentile 1%, che rappresenta la coda della distribuzione. Questa operazione ha rimosso alcune centinaia di certificazioni.

Multivariate outlier detection

La fase di *Data Mining* del *framework* aggrega tra di loro più attributi per effettuare *Clustering*. Anche se l'*outlier detection* univariata ha rimosso le anomalie sulle singole variabili, quando queste sono aggregate tra di loro, possono presentare dei valori non conformi. A questo proposito è stato necessario effettuare un'*outlier detection* multivariata, attraverso l'algoritmo *DBSCAN*. La Figura 3.3 mostra il K-Distance graph. Si può notare come la curva si stabilizzi ad un valore di MinPoints pari a 5. Il valore in corrispondenza dell'intersezione con la linea verde (il gomito) rappresenta il valore di Eps scelto, pari a 0.28. Con questi due parametri è stato avviato il DBSCAN, che ha permesso di isolare due areee, una a bassa densità, contenente circa un centinaio di certificati, identificativa di un *cluster* di *outlier*, e una seconda area, contenente il resto dei certificati del *dataset*.

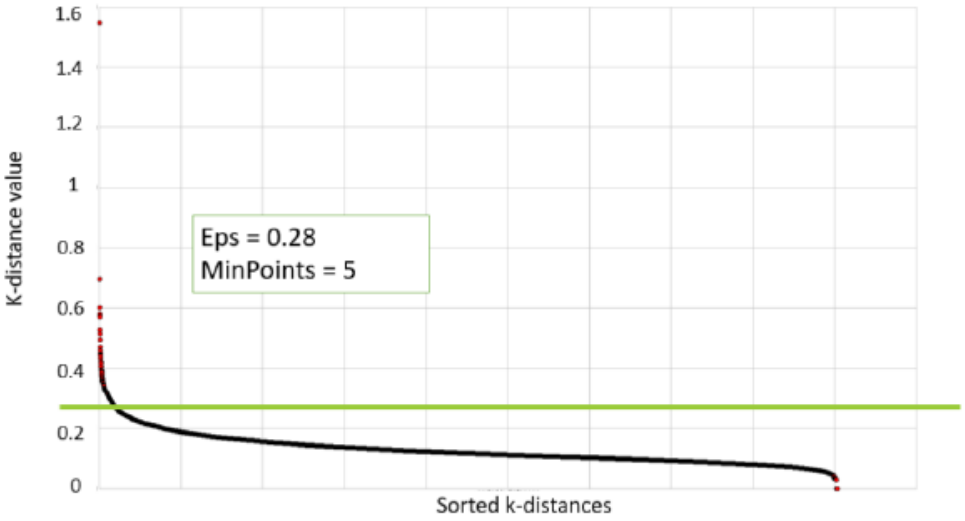

Figura 3.3: K-Distance graph per il settaggio dei parametri del DBSCAN

3.1.2 Data selection and Normalization

Data selection

È stata effettuata un'analisi di correlazione per supportare l'esperto di dominio nella scelta delle variabili significative su cui effettuare le prossime analisi. La *matrice di correlazione* in Figura 3.4 mostra i valori di correlazione di Pearson. Quando i valori sono prossimi a 1.0 significa che le variabili sono fortemente correlate. Si può notare come la superficie disperdente sia mediamente correlato con la superficie riscaldata, lo stesso vale per la superficie disperdente e il fattore forma. Occorre sottolineare come i rendimenti siano stati sostituiti dalla variabile ETAH, che tiene conto di questi contributi in maniera sintetica. Sono state in definitiva selezionate le seguenti 7 variabili:

- **Fattore forma**
- **Trasmittanza media delle superfici opache**
- **Trasmittanza media delle superfici trasparenti**
- **Superficie riscaldata**

- **Superficie disperdente**
- **Anno di costruzione**
- **Rendimento per la climatizzazione invernale ETAH**

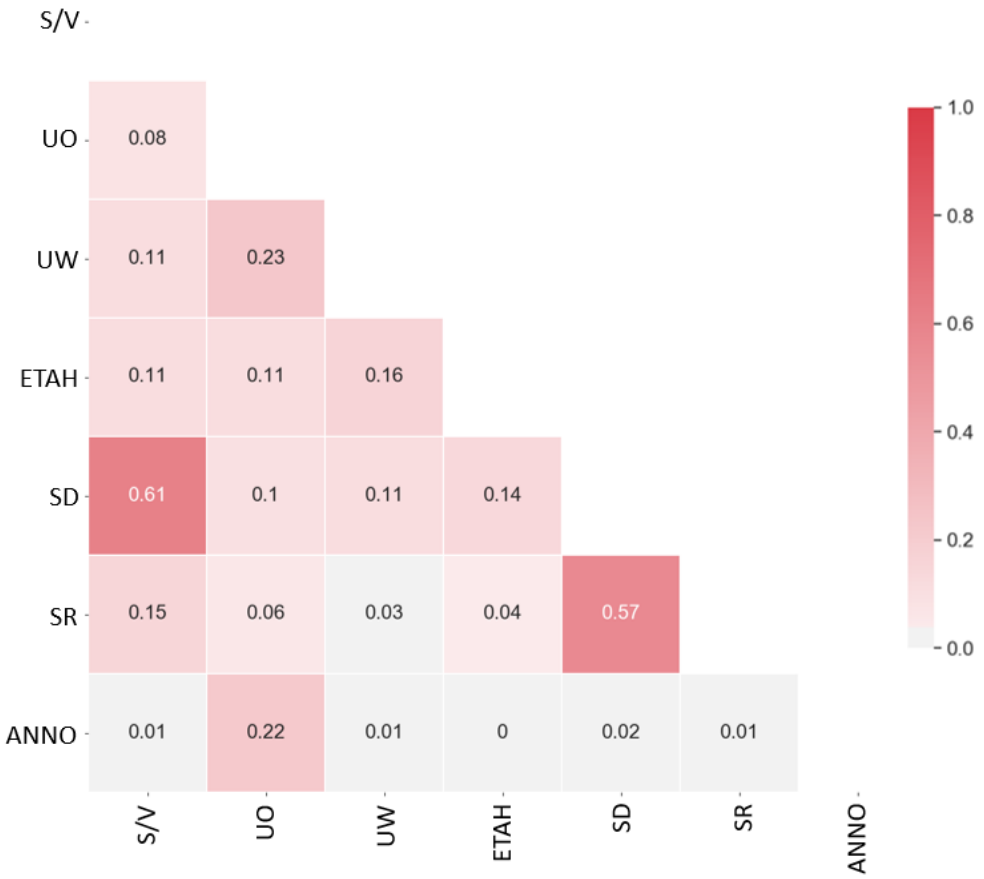

Figura 3.4: Matrice di correlazione usata per l'analisi

Normalization

È stata scelta la tecnica di normalizzazione *max-min*, con min pari a 0 e max uguale a 1, perchè mantiene la relazione nei dati originali. La normalizzazione è fondamentale, perchè la fase successiva del *framework*, ovvero l'applicazione dell'algoritmo di

Clustering K-Means, si basa proprio sul concetto di distanza.

3.2 Applicazione algoritmo di Clustering

Come algoritmo utilizzato per effettuare il *Clustering* è stato scelto il K-Means, perchè è un algoritmo veloce, molto utilizzato e semplice, con l'unico inconveniente legato alla scelta del parametro k, problema che però è stato superato dal *framework* TUCANA, in grado di restituire in modo automatico il valore ottimale di k. Viene di seguito descritto il *setting* dei parametri.

3.2.1 Setting dei parametri

Per applicare l'algoritmo K-Means è fondamentale settare il parametro k. È stato utilizzato l'*Elbow Method* per ottenere il migliore valore, con la determinazione automatica del valore k. Il valore di SSE è stato calcolato per diversi valori di k nel range tra 2 e 50, come mostrato in Figura 3.5. Il risultato ottenuto è k pari a 12.

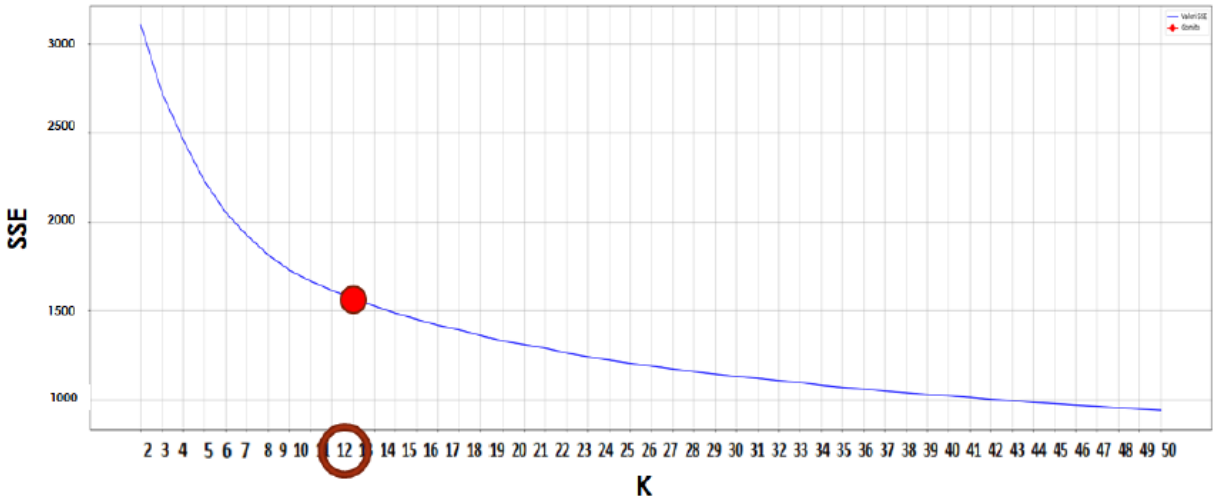

Figura 3.5: Elbow graph con K scelto automaticamente

3.2.2 Risultati K-Means

Successivamente è stato eseguito il K-Means con il valore di k uguale a 12, utilizzando come variabili di input quelle ottenute dalla fase di *Data Selection* normalizzate, e come tecnica per il calcolo della distanza quella euclidea. Il risultato del *Clustering* ha portato a 12 gruppi, la cui cardinalità è riassunta dalla Tabella 3.6. Si può constatare come i *cluster* 8 e 6 siano quelli più numerosi, mentre i *cluster* 3 e 10 quelli più piccoli.

Cluster ID	**% APE**
Cluster 0	5.9
Cluster 1	6.0
Cluster 2	5.6
Cluster 3	2.9
Cluster 4	9.0
Cluster 5	4.8
Cluster 6	13.6
Cluster 7	12.0
Cluster 8	16.0
Cluster 9	12.0
Cluster 10	2.7
Cluster 11	8.4

Tabella 3.6: Percentuale di certificati per i *cluster* ottenuti con k uguale a 12

La distribuzione dei certificati all'interno dei *cluster* rispetto alle circoscrizioni, mostrata in Tabella 3.7, ci fa notare come ogni gruppo abbia una percentuale variabile di certificati presenti in tutte le circoscrizioni, ad eccezione del *cluster* 10, la cui distribuzione non è uniforme all'interno della città di Torino, infatti esso rappresenta edifici storici (come si vedrà in seguito).

	Circostrizione							
Etichetta di cluster	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	0.34	0.82	1.10	0.72	0.94	0.74	0.57	0.75
1	0.77	0.96	1.03	0.83	0.43	0.45	0.48	1.05
2	0.30	0.79	0.88	0.94	0.87	0.37	0.65	0.80
3	0.83	0.28	0.30	0.26	0.07	0.14	0.36	0.69
4	0.73	1.3	1.7	1.0	1.0	0.90	0.97	1.4
5	1.43	0.61	0.78	0.55	0.11	0.12	0.35	0.87
6	1.3	2.5	2.3	1.6	1.3	1.0	1.2	2.5
7	1.39	1.44	2.12	1.60	1.38	2.08	1.17	1.71
8	1.5	2.5	2.9	2.2	2.0	1.5	1.7	2.3
9	2.60	1.97	2.14	1.57	1.17	0.91	1.19	1.85
10	2.1	0.0068	0.027	0.047	0.0033	0.030	0.18	0.26
11	0.85	1.07	1.46	0.81	1.00	0.89	0.97	1.34

Tabella 3.7: Cardinalità *cluster* (in %) per circoscrizione

3.2.3 Caratterizzazione cluster

Per la caratterizzazione dei risultati del *Clustering* sono stati utilizzati tre strumenti molto utili per l'esperto di dominio: *boxplot*, *radar chart* e *CART*. Nel dettaglio si vanno ad analizzare i *cluster* 0 e 4, attraverso l'utilizzo dei *boxplot*, un ottimo strumento di analisi, in grado di fornire informazioni precise sulla distribuzione delle sette variabili utilizzate per il *Clustering*, e il *radar chart*, rappresentativo del centroide del *cluster*. Come si può vedere dai *boxplot* e *radar chart* in Figura 3.6, il *cluster* 0 presenta un anno di costruzione recente, trasmittanze opache e trasparenti basse e un fattore forma contenuto, caratteristiche riconducibili ad edifici performanti.

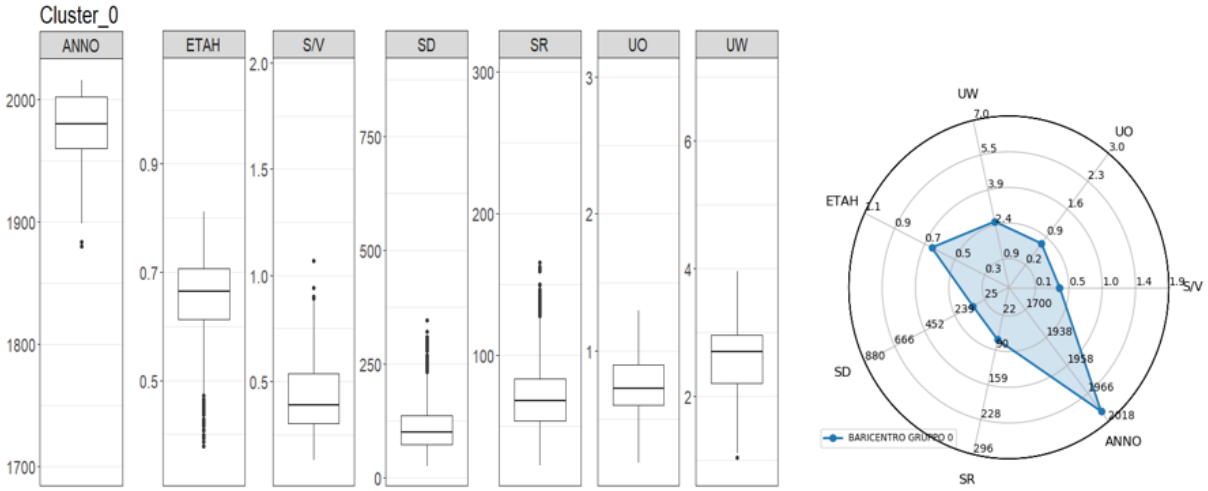

Figura 3.6: Boxplot e *radar chart* per il *cluster* 0

Il *cluster* 4 è rappresentativo di edifici con un anno di costruzione vecchio, trasmittanze opache e trasparenti alte e un fattore forma poco contenuto, come si nota in Figura 3.7, proprietà di edifici poco efficienti.

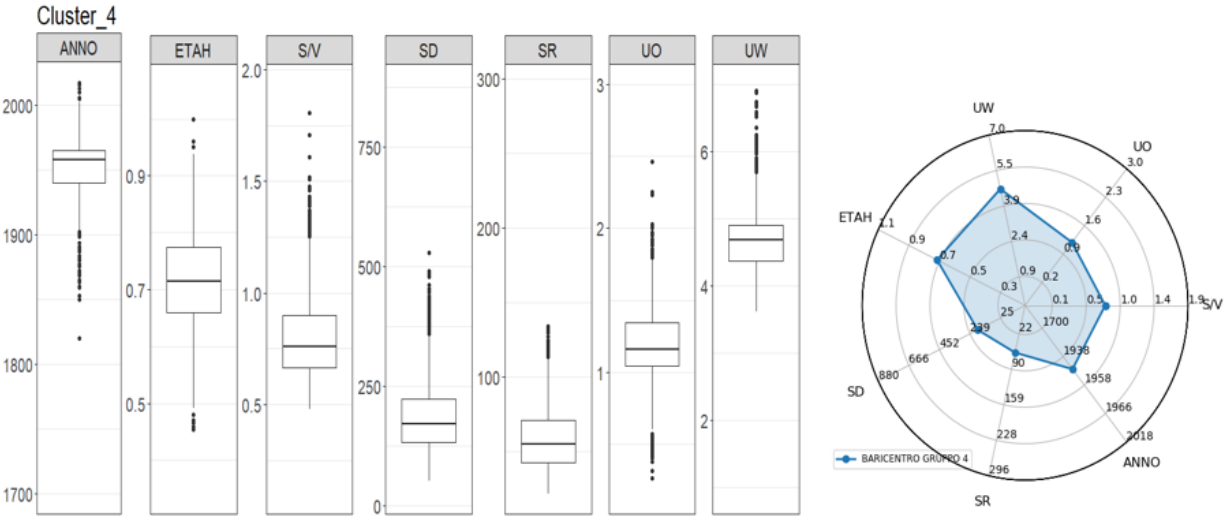

Figura 3.7: Boxplot e *radar chart* per il *cluster* 4

Confrontando due gruppi di edifici con performance basse (Figure 3.7 e 3.8), come

i *cluster* 4 e 11, si può constatare come essi presentino differenze in almeno una variabile, ovvero il *cluster* 4 è caratterizzato da trasmittanza più alta rispetto al *cluster* 11.

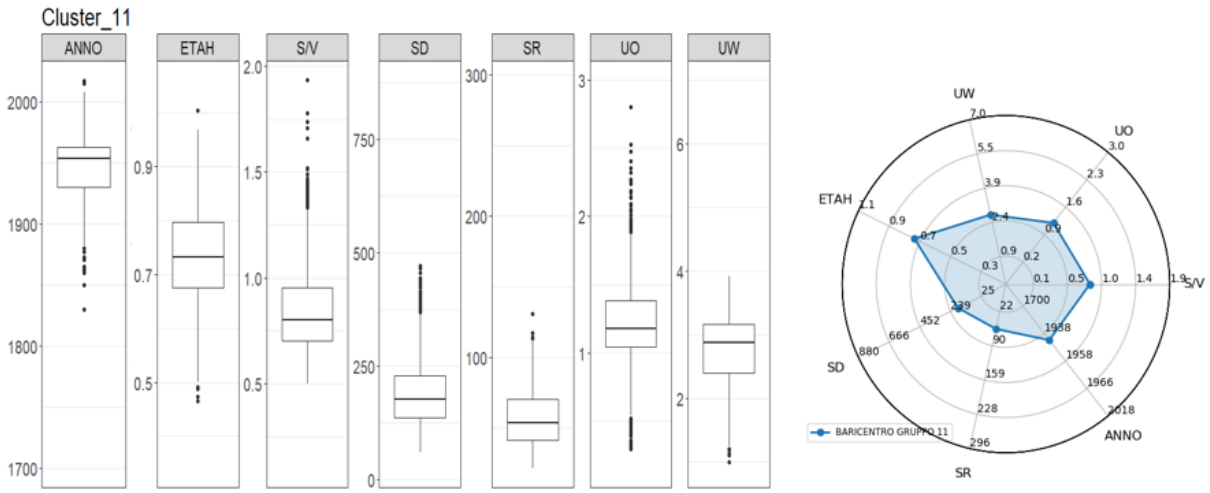

Figura 3.8: Boxplot e *radar chart* per il *cluster* 11

Paragonando invece due gruppi di edifici con prestazioni energetiche alte (Figure 3.6 e 3.9), come i *cluster* 0 e 2, è possibile appurare come il *cluster* 0 presenti ETAH più basso, superficie riscaldata più bassa e trasmittanza opaca più alta rispetto al *cluster* 2. In definitiva, seppur dei *cluster* abbiano una prestazione energetica comune, sono parte di gruppi distinti.

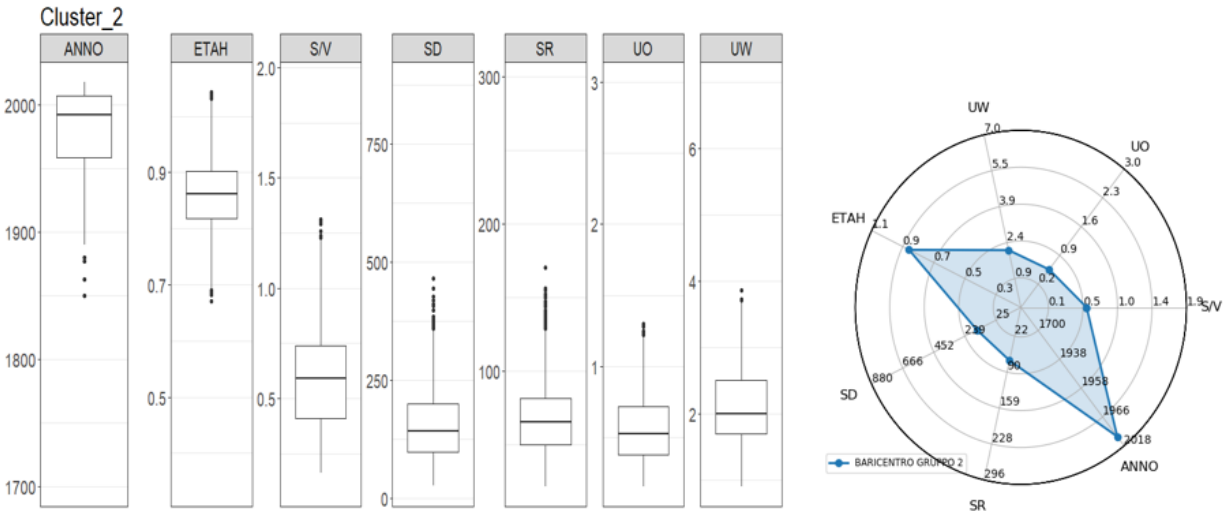

Figura 3.9: Boxplot e *radar chart* per il cluster 2

Con i *boxplot* e *radar chart* è possibile analizzare singolarmente le variabili di *Clustering*. Con il CART l'esperto di dominio può invece analizzare le variabili congiuntamente e caratterizzare i gruppi generati dall'algoritmo di *Clustering* attraverso l'estrazione di regole *IF-THEN*. Esso è stato costruito utilizzando la funzione *rpart* in R, impostando il *Complexity Parameter* a 0.00098, il minimo numero di punti per nodo foglio a 300, la profondità massima dell'albero a 5, e analizzando l'andamento in funzione dell'errore relativo commesso, attuando una 10-*fold cross-validation*. In Figura 3.10 è raffigurato una parte dell'albero di decisione, mentre nella Tabella 3.8 sono riportate alcune regole estratte.

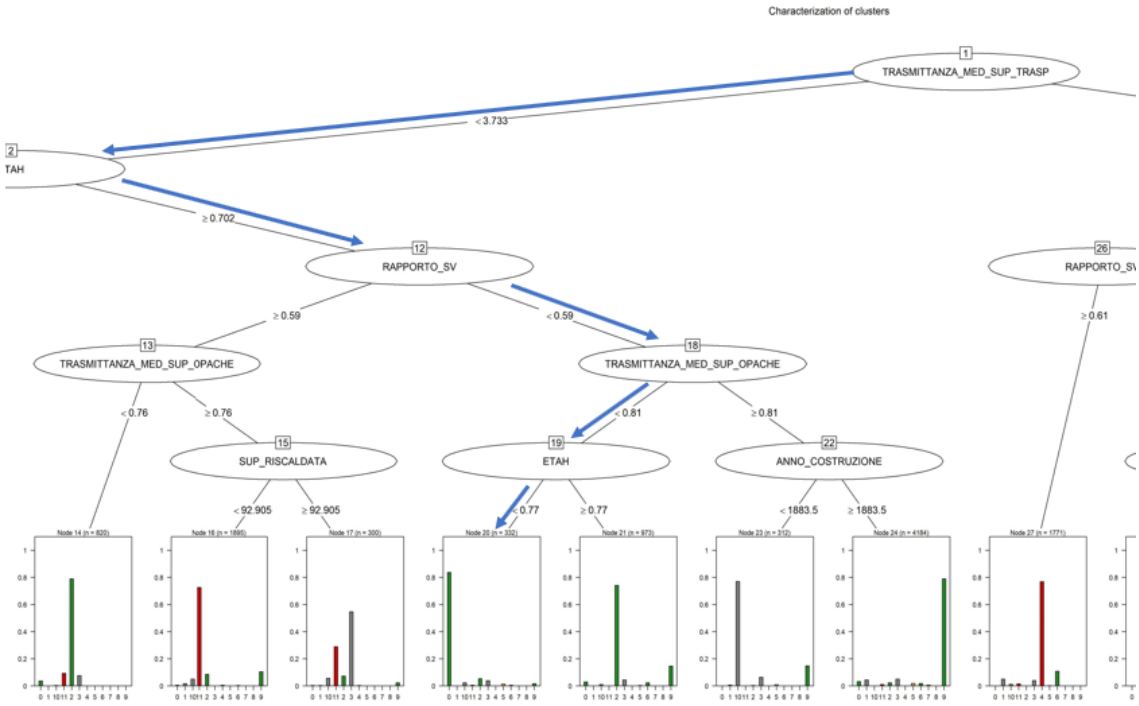

Figura 3.10: Dettaglio del CART generato per la caratterizzazione dell'etichetta di *Clustering*

Si è ottenuto un valore di circa l'80% per l'accuratezza del CART, che dimostra come esso possa essere considerato robusto per la caratterizzazione ottenuta.

IF	**THEN**
TRASM. TRASP $<$ 3.733 & ETAH $\geq$ 0.702 & SV $<$0.59 & TRAM. OPACA $<$ 0.81 & ETAH $<$ 0.77	ClusterID = 0
TRASM. TRASP $\geq$ 3.733 & ETAH $\leq$ 0.699 & *SV* $\geq$ 0.61	ClusterID = 4
TRASM. TRASP $\geq$ 3.733 & ETAH $\leq$ 0.699 & SV $<$0.61 & TRAM. OPACA $\geq$ 1.46 & ETAH $<$ 0.814	ClusterID = 1
TRASM. TRASP $\geq$ 3.733 & ETAH $\leq$ 0.699 & SV $<$0.61 & TRAM. OPACA $\geq$ 1.46 & ETAH $\geq$ 0.814	ClusterID = 6

Tabella 3.8: Alcune regole estratte dal CART

Per completare la fase di *analytics*, analizzando i *boxplot*, i *radar chart* e le regole estratte dal CART, l'esperto di dominio ha assegnato delle etichette ai dodici *cluster* (Tabella 3.9), fornendo per ciascuna di esse una descrizione. L'etichetta 'Non Classificabile' è stata data a quei gruppi non omogenei in termini di efficienza energetica, le restanti tre invece sono state date a quei *cluster* rappresentativi in modo conforme di una determinata performance energetica.

Label	Performance	Descrizione
0	Alta	Involucro Performante, Impianto Mediamente Performante
1	Non Classificabile	Involucro Scarso, Fattore Forma Basso
2	Alta	Involucro e Impianto Performanti
3	Non Classificabile	Edifici con Ampia Metratura
4	Bassa	Involucro Scarso, Fattore Forma Alto
5	Media	Involucro Scarso, Impianto Mediamente Performante, Fattore Forma Basso
6	Alta	Involucro Scarso, Impianto Performante, Fattore Forma Basso
7	Media	Involucro Performante, Impianto Scarso, Fattore Forma Basso
8	Media	Involucro Mediamente Performante, Impianto Scarso, Fattore Forma Basso
9	Alta	Involucro e Impianto Mediamente Performanti, Fattore Forma Basso
10	Non Classificabile	Edifici Storici
11	Bassa	Involucro e Impianto Mediamente Performanti, Fattore Forma Alto

Tabella 3.9: Classificazione e caratterizzazione dei gruppi di edifici

3.3 Applicazione Web TUCANA

L'applicazione web TUCANA, è stata sviluppata, con lo scopo di visualizzare possibili pattern relativi alle certificazioni energetiche della città di Torino. Essa fornisce strumenti per effettuare alcune statistiche sulle prestazioni energetiche degli edifici e la visualizzazione interattiva, attraverso mappe geolocalizzate, della conoscenza estratta. L'applicazione web è stata pensata per tre tipologie di utenti:

- **Cittadino**: potrebbe essere utile per il cittadino acquisire informazioni sull'efficienza energetica della propria zona abitativa o della propria residenza, oltre a poter individuare delle zone in cui è conveniente acquistare o affittare un immobile nella città di Torino.
- **Pubblica amministrazione**: che vorrebbe effettuare interventi migliorativi in alcune zone della città.
- **Fornitore di servizi energetici**: che desidererebbero approfondire le caratteristiche della prestazione energetica degli edifici, al fine di individuare le zone della città in cui proporre soluzioni adatte alla tipologia di edifici presenti.

L'utente indicando la tipologia di appartenenza (vedi Figura 3.11), può accedere a determinati contenuti personalizzati erogati dall'applicazione web.

Figura 3.11: Schermata tipologia di utente dell'applicazione web

Sessioni personalizzate

Il meccanismo delle sessioni è utilizzato dall'applicazione web per selezionare la tipologia di utente desiderata e consentire una visione personalizzata di contenuti. Essendo il protocollo HTTP *stateless*, ogni richiesta è indipendente dalle altre, per questo motivo, grazie alle sessioni, il *server* può riconoscere richieste collegate tra di loro. Una sessione è avviata dal *server*, che crea un ID di sessione univoco, mantenendo le informazioni di sessione associate ad esso. In ogni richiesta che il client fa che deve essere riconosciuta come appartenente alla sessione, esso include l'ID di sessione. La sessione tipicamente scade in fase di logout o alla scadenza di un certo tempo. Il modo più diffuso per implementare le sessioni è tramite i *cookie*: il *server* invia l'ID tramite un *cookie* al client che lo immagazina. Il *cookie* viene incluso ad ogni richiesta che il client effettua. l'ID di sessione deve essere scelto in modo random dal sistema o dal programmatore, in modo che non sia facilmente indovinabile. Il *server* registra le informazioni raccolte durante la navigazione su un file di testo che verrà salvato in remoto, avente come nome l'ID di sessione, così che ogni utente avrà il propio file con le proprie variabili di sessione. Quindi, mentre i *cookie* sono salvati sul pc dell'utente, le sessioni salvano i dati sul *server* stesso.
In *Flask* le sessioni sono gestite tramite l'estensione *session*. Per esempio, per settare la variabile di sessione 'cittadino' si usa l'istruzione: session['user'] = 'cittadino'. Invece per rilasciare una variabile di sessione si usa il metodo *pop()*: session.pop('user', None). I *cookie* di sessione sono firmati crittograficamente dal *server*, per questo occorre definire una SECRET_KEY. Quindi nessuno può modificare il *cookie* di

sessione, a meno che non possieda la chiave segreta. Ogni richiesta al *server*, una volta che che il *cookie* di sessione è settato, verifica l'autenticità del coofie utilizzando la chiave segreta. Se *Flask* non riesce a firmare il *cookie*, il suo contenuto viene scartato e un nuovo *cookie* di sessione viene inviato al browser. Nel *template* per verificare che la tipologia di utente sia, ad esempio, quella del "Cittadino", si effetua il seguente controllo: {% if session['user'] != 'Cittadino' %}.

Mappe

Le tipologie di mappe presentate nell'applicazione web, sono quelle presentate nel paragrafo 2.3.2: mappe coropletiche, mappe *scatter* e mappe *marker-cluster*. Esse sono state realizzate attraverso la libreria Folium, che sfrutta i punti di foza dell'ecosistema Python per la manipolazione dei dati, e quelli della libreria Leaflet di Javascript, per la parte di visualizzazione. Folium presenta una varietà di tileset incorporati, in particolare per questa applicazione web è stato utilizzato il tileset gratuito *OpenStreetMap*. L'output prodotto da questa libreria è un file HTML, che viene ancorato nella pagina HTML padre tramite un *iframe*. Per creare una mappa occorre utilizzare la funzione folium.Map(), che prende come parametri le coordinate dell'area geografica da visualizzare, lo zoom di partenza, lo zoom massimo consentito e il tileset.

Mappe Coropletiche

L'applicazione web dispone la visualizzazione di mappe coropletiche basate sulla scelta di uno specifico attributo energetico, presente all'interno della base dati degli attestati energetici della città di Torino: ETAH, Fattore forma, Anno di costruzione, Trasmittanza media delle superifici opache, Trasmittanza media delle superfici trasparenti, Superficie disperdente totale e Superficie riscaldata. In Folium le mappe coropletiche vengono create utilizzando la funzione Map.choropleth(), specificando i valori limite per la scala di colori da utilizzare e facendo un collegamento tra i dati presenti in un DataFrame Pandas e il contenuto di un file GeoJSON, contenente una collezione di poligoni. Le mappe coropletiche realizzate vengono quindi suddivise in poligoni che si colorano tramite il risultato di un *Majority Model* per un determinato attributo scelto, ovvero quando la maggioranza di certificati energetici presenti in un poligono prende valore in un certo intervallo, allora, esso fa riferimento al colore compreso tra gli estremi dell'intervallo considerato. Inoltre, quando il numero

di certificati presenti all'interno di un dato poligono risulta essere inferiore a 3, il poligono si colora di grigio. Quando invece, non sono presenti attestati energetici in un determinato poligono, esso si colora di grigio scuro. Un esempio di mappa coropletica per l'attributo ETAH è visibile in Figura 3.12.

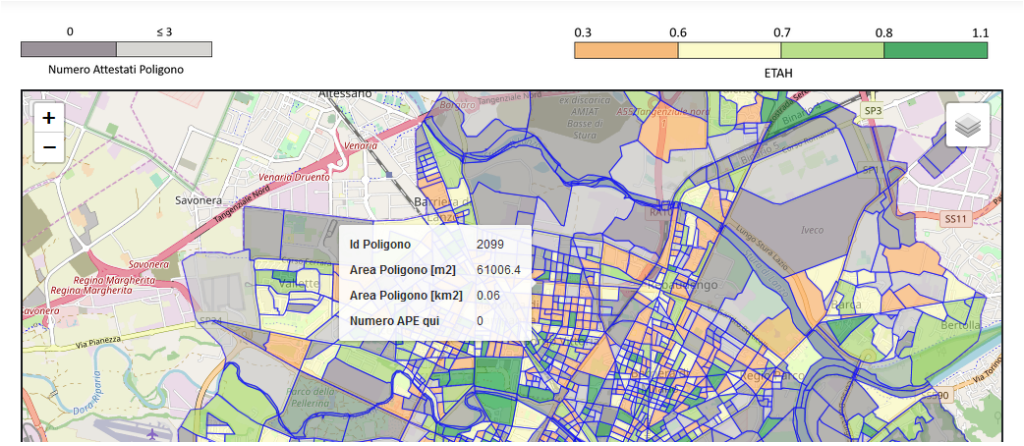

Figura 3.12: Mappa coropletica per l'attributo energetico ETAH

Per le tipologie di utenti: Pubblica Amministrazione e Fornitore Servizi Energetici, è consentito visionare una tipologia di mappa coropletica aggiuntiva, di scarsa comprensione e utilità per la tipologia 'Cittadino'. Tale mappa è sempre suddivisa in poligoni che si colorano tramite l'applicazione del *Majority Model*, ma questa volta sul risultato ottenuto dalle analisi di *Clustering*. Un esempio di tale tipologia di mappa è visibile in Figura 3.13. In particolare, la performance energetica della zona delimitata da un poligono, è definita dalla prestazione energetica dei *cluster* che formano la maggioranza all'interno del poligono. Quando il numero di attestati energetici presenti all'interno di un certo poligono risulta essere inferiore o uguale a 3, allora il poligono non viene colorato. Questo perchè non si ha un numero di certificati sufficienti per poter classificare dal punto di vista energetico un certo isolato. Passando il mouse sopra le mappe coropletiche analizzate è possibile visionare, attraverso un tooltip, le informazioni seguenti: ID, area e numero di attestati energetici presenti nel poligono.

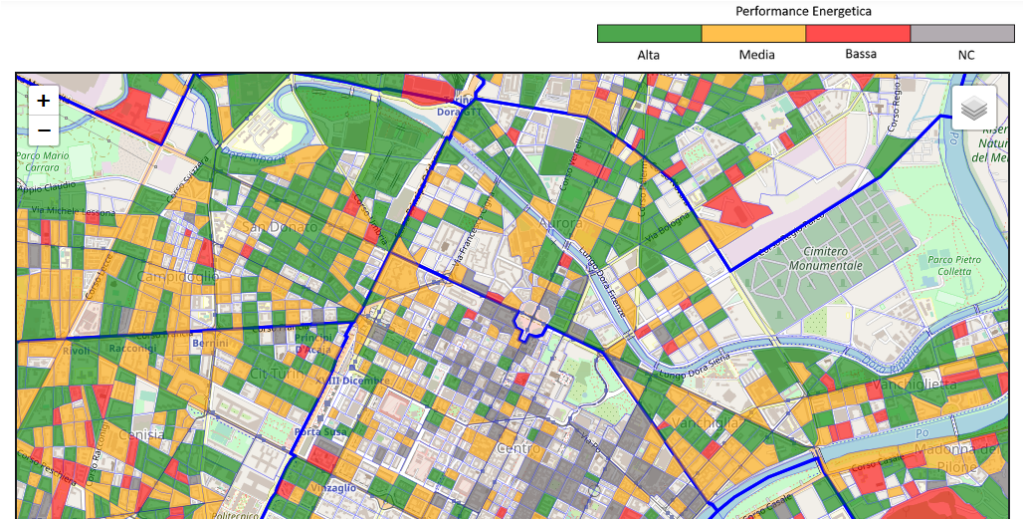

Figura 3.13: Mappa coropletica per il risultato della combinazione di 7 attributi

Mappe Scatter

Le mappe *scatter* proposte dall'applicazione web permettono di visualizzare dati geografici attraverso dei *marker*. In Folium i marker vengono realizzati con la funzione folium.Marker(), di cui vengono passate le coordinate di latitudine e longitudine, un popup e l'icona (realizzata attraverso il plugin BeautifyIcon). Il colore dei marker viene assegnato in base all'etichetta di *Clustering*. Ogni marker rappresenta una singola unità abitativa a cui è associato un attestato di prestazione energetica. Tale visualizzazione è molto utile in tutti quei casi in cui si voglia concentrare l'analisi su specifici edifici. La numerosità di APE presenti nel *dataset* non permette una visione d'insieme agevole, perchè un numero troppo grande di marker rallenterebbe sia il caricamento della mappa che la sua navigazione da parte dell'utente. Per questo motivo si è data la possibilità di scegliere tra diversi metodi di visualizzazione:

- **Un certificato specifico**: tale modalità permette di visualizzare un edificio in particolare e alcuni dei suoi vicini, nel caso in cui l'utente conosca l'indirizzo, oppure i dati catastali riferiti all'edificio cercato.
- **Un gruppo di certificati**: Indicando un livello di performance energetica (Alta, Media, Bassa, Non classificabile), l'identificativo di una circoscrizione

e il numero di attestati APE che si intendono visualizzare, questa modalità consente all'utente di conoscere alcune variabili energetiche degli edifici con il livello di performance scelto.

Andando ad analizzare il primo metodo di visualizzazione, l'utente ha la possibilità di indicare un'opzione tra:

- **Edificio - Indirizzo**: indicando l'indirizzo di un edificio, l'Id identificativo della sua circoscrizione e il numero massimo di vicini che si intende visualizzare, questa opzione permette all'utente di conoscere alcune variabili energetiche dell'edificio di interesse e del suo vicinato. Nel caso in cui l'indirizzo inserito dall'utente non sia presente all'interno della base dati, si distinguono due possibili risultati: se la via inserita non è presente all'interno del *dataset*, allora viene stampato un messaggio di errore; se la via inserita dall'utente è presente, ma il numero civico inserito non trova alcuna corrispondenza, allora viene mostrato sulla mappa l'edificio (ed il suo vicinato) con il civico più vicino a quello inserito dall'utente. Un esempio di questa tipologia di mappa è raffigurata in Figura 3.14.

- **Edificio - Dati Catastali**: indicando i dati catastali di un edificio: foglio, particella e subalterno, e il numero massimo di vicini che si intende visualizzare, questa opzione permette all'utente di conoscere alcune variabili energetiche dell'edificio di interesse e del suo vicinato. Nel caso in cui i dati catastali inseriti dall'utente non siano presenti all'interno della base dati degli attestati energetici, allora il sito non visualizza la mappa e viene stampato a video un messaggio di errore.

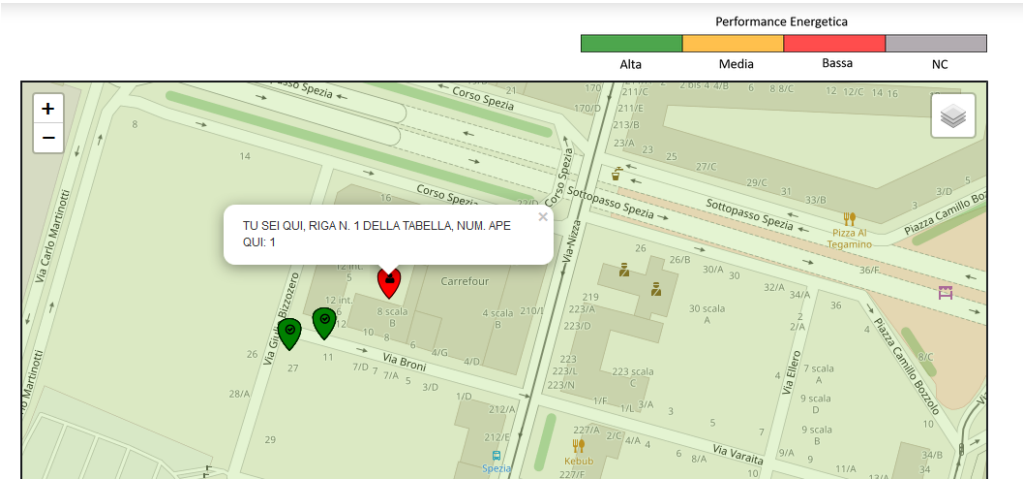

Figura 3.14: Mappa *scatter* edificio - indirizzo

All'interno del *dataset* sono presenti più attestati energetici di unità abitative appartenenti allo stesso edificio, per questo motivo il popup dei marker mostra il numero di attestati in quel punto. In aggiunta alle mappa visualizzate, viene presentata anche una tabella riassuntiva con alcune informazioni energetiche per ogni abitazione che viene raffigurata sulla mappa, da poter scaricare in formato csv (comma-separated values).

Mappe Marker-Cluster

Per le tipologie di utenti: Pubblica Amministrazione e Fornitore Servizi Energetici, si è prevista la possibilità di visionare delle mappe *marker-cluster*. Tali mappe sono in grado di fornire una visione aggregata degli attestati APE e sono il risultato di un'analisi effettuata sui 7 attributi energetici. Le abitazioni con caratteristiche termo-fisiche simili sono raggruppate in un cerchio ad un livello di zoom meno dettagliato. Per realizzare questa tipologia di mappa si è utilizzato un plugin della libreria Folium, noto come *marker-cluster*. La modifica al comportamento personalizzato dei *marker-cluster* è data dalla funzione *iconcreatefunction()*, che viene passata nel costruttore dell'oggetto e che da la possibilità di personalizzare il colore dei *cluster* e la loro dimensione. In particolare, tale funzione è stata modificata settando cinque possibili dimensioni dei *marker-cluster* in base al numero di certificati energetici presenti al loro interno, impostando come colore quello dell'etichetta di *Clustering* e indicando il numero di attestati energetici presenti al loro interno. Un

esempio di mappa *marker-cluster* con performance energetica media viene mostrata di seguito:

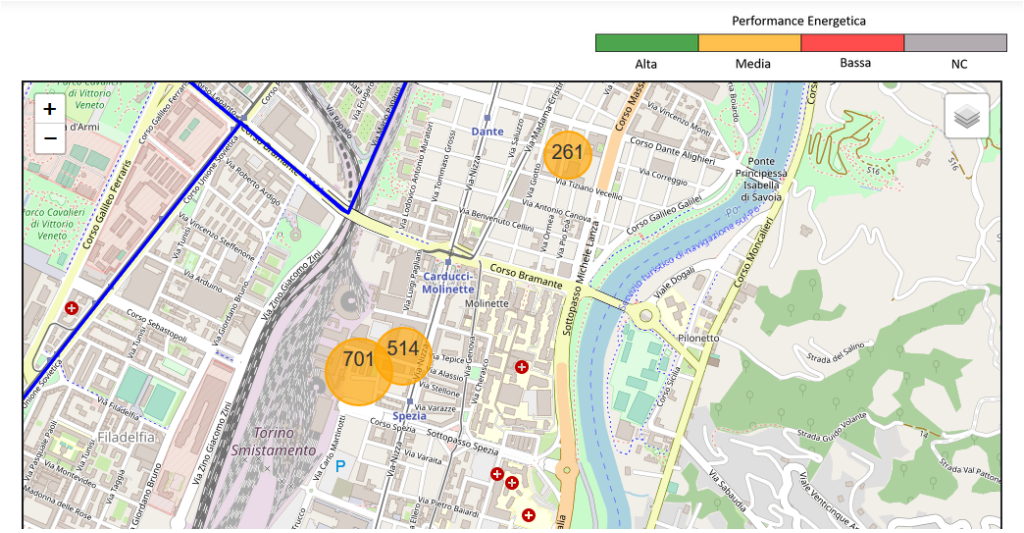

Figura 3.15: Mappa *marker-cluster* performance media

Oltre alla mappa geolocalizzata, nella pagina web dedicata alle mappe *marker-cluster*, vengono fornite alcune statistiche rappresentate mediante grafici opportuni, ovvero:

- Un *radar chart* che mette a confronto il baricentro del gruppo nella circoscrizione selezionata con il baricentro del gruppo pensato su tutto il *dataset*.
- Un boxplot che evidenzia la distribuzione delle 7 variabili di analisi.
- La distribuzione dell'Indice di Prestazione Energetica (EPH).

Tali statistiche vengono visualizzate quando si clicca sul *marker-cluster*, facendo comunicare l'iframe con la pagina HTML padre. In particolare l'iframe invia un messaggio alla finestra padre utilizzando il metodo *window.parent.postMessage*, che permette la comunicazione sicura tra di essi. Il primo parametro ricevuto dal metodo è il messaggio che nel nostro caso rappresenta le immagini dei grafici sopra citati (vedi Figura 3.16). Il secondo parametro specifica quale deve essere l'origine della finestra padre per l'evento da inviare, sia come stringa letterale "*" (che indica alcuna preferenza) che come URI. Se quando l'evento è pianificato per la spedizione,

il protocollo, il nome dell'host e la porta del documento HTML padre non corrispondono con quello previsto da tale parametro, l'evento non verrà inviato. Questo meccaniscmo fornisce un controllo su dove vengono inviati i messaggi. La finestra padre rimane in ascolto attraverso il metodo *window.addEventListener()*, che alla ricezione dell'evento fa scaturire una funzione che innesta nel documento HTML il messaggio ricevuto.

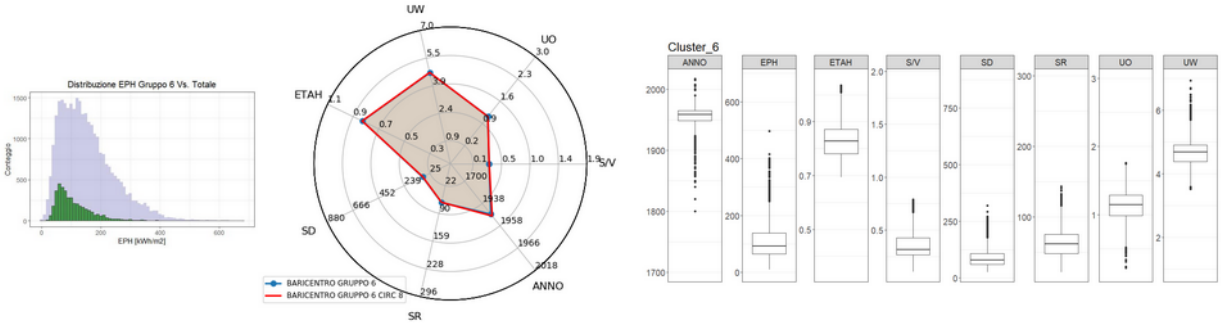

Figura 3.16: Statistiche relative al *cluster* 6

Statistiche

L'applicazione web sviluppata, con l'intento di guidare l'utente verso un'approfondita conoscenza delle caratteristiche degli edifici residenziali della città di Torino, presenta una sezione apposita per la consultazione di statistiche per ogni attributo energetico utilizzato per le analisi dei dati, e pochi altri utilizzati per la compilazione degli APE. In particolare, possono essere visualizzati alcuni diagrammi tipici della statistica descrittiva, come diagrammi a torta e istogrammi, e tabelle. Andando a prendere in considerazione le statistiche riferite all'anno di costruzione, si può vedere in Figura 3.17 un *bar chart* realizzato con la libreria *Matplotlib*. Si può constatare una grande presenza di certificati tra gli anni 50 e 70. Il motivo dell'ampia numerosità di certificati per gli edifici di questi anni può essere dovuto al fatto che, prima di questo periodo, non esistevano obblighi legislativi nell'ambito energetico.

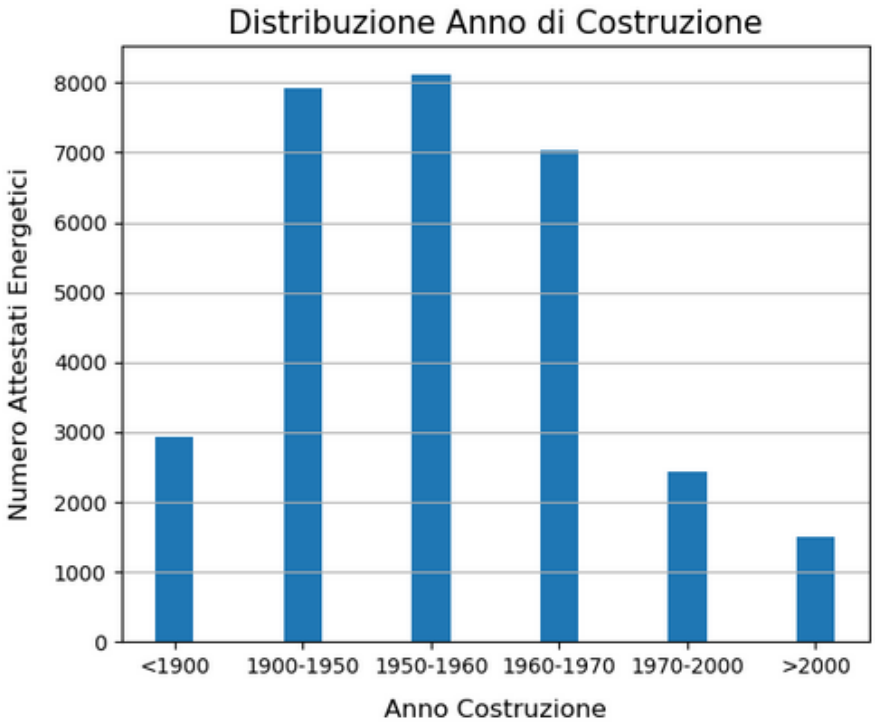

Figura 3.17: Bar chart per l'anno di costruzione

Validazione degli input

I form presenti nell'applicazione web sono stati validati sia lato *front-end* che *back-end*. Nel primo caso è stato utilizzato Javascript per effettuare dei controlli sulla correttezza dei campi. In particolare è stato verificato che i campi non fossero lasciati vuoti e che il formato fosse corretto, attraverso l'utilizzo delle espressioni regolari. In caso di mancata o errata immissione, viene visualizzato un *alert* che avvisa l'utente dell'errore presentatosi. Lato *back-end* è stata utilizzata l'estensione Flask-WTF, che fornisce un'interfaccia a WTForms, una libreria flessibile di validazione e rendering. Flask-WTF utilizza le classi per rappresentare i form e la SECRET_KEY per proteggere i form dagli attacchi di CSRF (*Cross-Site Request Forgery*).

3.4 Generalizzazione della conoscenza

La generalizzazione della conoscenza ha come obiettivo il prevedere il comportamento energetico degli edifici non presenti nel *dataset* utilizzato. Il *framework* realizzato permette di gestire due possibili tipi di predizioni:

- **Variabili di analisi mancanti**: caso in cui la nuova certificazione non ha una delle sette variabili di *Clustering*.
- **Classificazione**: situazione in cui il nuovo attestato energetico ha, o tutte e tre le variabili geometriche, oppure tutti gli attributi.

I percorsi di generalizzazione sono due, quello completo permette di predirre alcune variabili mancanti fino all'etichettatura di *Clustering*, quello semi-completo prende in considerazione la sola classificazione degli edifici tramite etichettatura di *Clustering*.
Si è scelto di confrontare la tecnica *K-fold cross validation* con metodologia *leave-one-out cross validation* al fine di valutare le performance dei modelli di regressione e classificazione. Partendo da un *dataset* di circa 30.000 certificati ed effettuando alcuni esperimenti, si è arrivati alla scelta di un K pari a 3. Il modello semi-completo considera, per la *leave-one-out cross validation* il 75% dei certificati come *training set* e il 25% come *test set*. Per il modello completo sono stati presi l'80% dei dati come *training set* e il restante 20% come *test set*. Entrambe le metodologie hanno portato a risultati comparabili, si è allora scelta la *leave-one-out cross validation* per il suo carico computazione più basso.
Come fase preliminare viene effettuata la normalizzazione del *training set* e *test set*. Per il primo viene attuata la *Z-score*, che considera la deviazione standard e la media del *training set*. Per il *test set* si applica la stessa tecnica ma la media e la deviazione standard sono riferite ai dati di *training set*

3.4.1 Realizzazione del modello Semi-Completo

La predizione dell'etichetta di *cluster* di un nuovo edificio, avviene o quando il certificato energetico presenta tutte e sette le variabili di *Clustering*, o quando ha solamente le tre variabili geometriche. Il modello semi-completo implementa l'algoritmo *K-Nearest Neighbors*.

Primo caso: presenza di tutte le variabili

Quando si hanno tutte le variabili di *Clustering*, si procede in due passi:

- **Vicinato geografico**: Si considerano gli attributi spaziali, ovvero latitudine e longitudine, e si calcola la distanza euclidea tra il dato di test e un certo numero di dati di *training*.

- **Similarità K-NN**: il secondo passo applica l'algoritmo K-NN per considerare solo le osservazioni del *training set* più vicine al dato di test. Si applica la distanza euclidea tra tutte le variabili del test e quelle del suo vicinato geografico.

Si prende l'etichetta che si presenta più volte all'interno del vicinato selezionato dal K-NN per fare la predizione della *label*. La scelta del numero di osservazioni e del parametro K, viene ottenuta con la *grid search*, che prende diverse combinazioni per questi due parametri, misurandone l'accuratezza. In Figura 3.18 sono riportati i valori di accuratezza per diverse combinazioni. Scegliendo un vicinato pari a 1000 e K pari a 50 si ottiene un'accuratezza pari a 0.837. Il valore più alto di accuratezza si ottiene per un vicinato pari a 2000 e K sempre uguale a 50, tuttavia il guadagno in questo caso è minimo in confronto al numero di osservazioni.

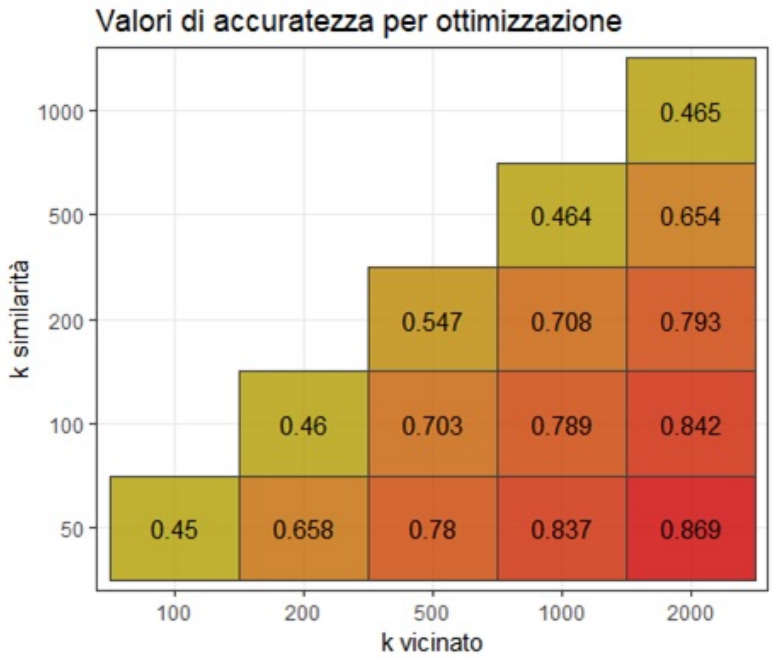

Figura 3.18: Tecnica grid-search per il K-NN con tutte le variabili presenti

In Tabella 3.10 sono riportati i valori medi di precisione e richiamo per l'accuratezza ottenuta.

Accuratezza	**Precisione Media**	**Richiamo Medio**
0.837	0.876	0.761

Tabella 3.10: Valori di precisione e richiamo medio con tutte le variabili di analisi.

Invece in Tabella 3.11 sono riportati i valori di precisione e richiamo per ogni etichetta di *cluster*.

Label	Precisione	Richiamo
0	0.917	0.576
1	0.951	0.480
2	0.942	0.662
3	0.962	0.481
4	0.897	0.861
5	0.820	0.580
6	0.839	0.950
7	0.787	0.950
8	0.842	0.989
9	0.765	0.954
10	0.963	0.792
11	0.829	0.863

Tabella 3.11: Valori di precisione e richiamo per ogni etichetta di *cluster* con tutte le variabili di analisi

Secondo caso: presenza delle sole variabili geometriche

Quando invece si hanno le sole variabili geometriche, ovvero: fattore forma, superficie riscaldata e superficie disperdente, si esegue, come nel caso precedente, la tecnica a due step, avendo però meno variabili per l'etichettatura di *Clustering*. I risultati del *grid-search* sono riportati figura:

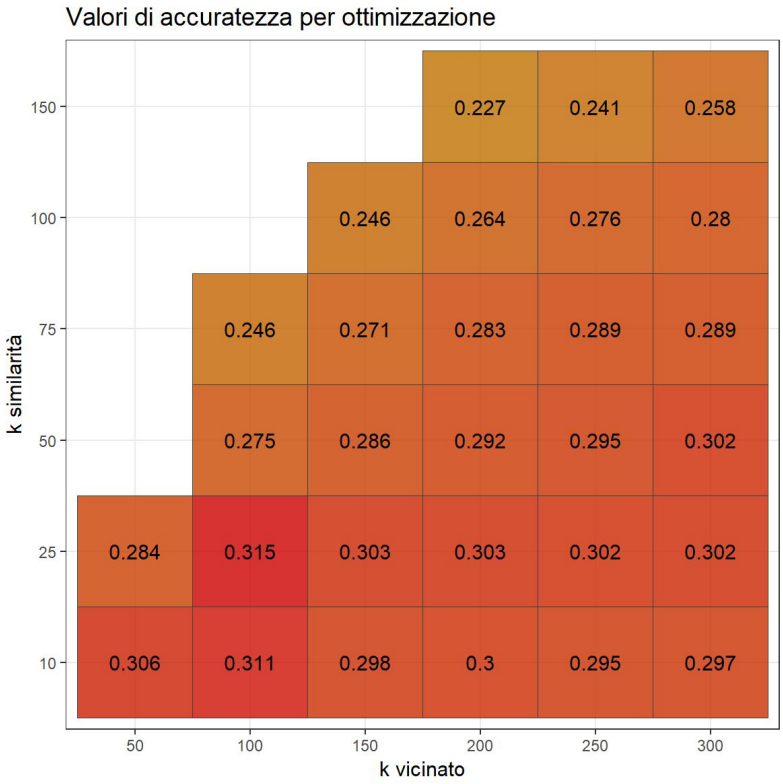

Figura 3.19: Tecnica *grid-search* per il K-NN con le sole variabili geometriche

I valori di accuratezza sono molto più bassi rispetto al caso precedente, questo perchè avendo solo tre variabili a disposizione, la classificazione diventa più difficoltosa. Si sono scelti un numero di vicini pari a 100 e un K pari a 25, corrispondenti all'accuratezza più alta. La Tabella 3.12 riporta i valori di precisione e richiamo medi per l'accuratezza ottenuta.

Accuratezza	**Precisione Media**	**Richiamo Medio**
0.315	0.343	0.311

Tabella 3.12: Valori di precisione e richiamo medio per il modello con le sole variabili geometriche

La Tabella 3.13 riporta i valori di precisione e richiamo per ogni etichetta di *cluster*.

Label	Precisione	Richiamo
0	0.292	0.204
1	0.117	0.051
2	0.319	0.261
3	0.790	0.299
4	0.358	0.415
5	0.497	0.467
6	0.245	0.306
7	0.247	0.185
8	0.274	0.413
9	0.203	0.142
10	0.381	0.564
11	0.397	0.429

Tabella 3.13: Valori di precisione e richiamo per ogni etichetta di *cluster* con le sole variabili geometriche

3.4.2 Realizzazione del modello Completo

Anche per il modello completo, la fase preliminare riguarda la normalizzazione dei dati con la tecnica *Z-score*. Tale modello ha come obiettivo la predizione di alcuni attributi mancanti e la successiva predizione della *label* di *Clustering*, con la metodologia vista in precedenza. Si considerano i casi in cui manchino o il fattore forma o l'ETAH. Per entrambi viene applicata la regressione per la predizione dei valori.

Primo caso: ETAH mancante

L'ETAH, come già visto nel primo capitolo, è dato dal prodotto dei quattro rendimenti energetici: ETAG, ETAE, ETAR e ETAD. Il *framework* permette di predirre l'ETAH nei casi in cui, dato un nuovo certificato in arrivo, uno fra i quattro rendimenti sopra citati manchi. Per ottenere un modello di regressione performante, si somma, alle variabili analisi, anche il prodotto dei rendimenti presenti nel dato di test. Se ad un'osservazione di test manca, per esempio, l'attributo ETAG, si genera il modello di regressione sotto riportato:

$$ETAH \sim S/V + UW + UO + SR + SD + ANNO + ETAE * ETAR * ETAD\,.$$

I modelli di regressione implementati sono quelli lineare, polinomiale, lasso e K-NN. Sono state effettuate due *grid-search*, una per la scelta di K per il K-NN (Tabella 3.14) e l'altra per la determinazione del parametro di *tuning* λ (Tabella 3.15).

K	R^2 ETAD	R^2 ETAE	R^2 ETAG	R^2 ETAR
1	0.82	0.92	0.55	0.42
3	0.89	0.94	0.68	0.60
5	0.89	0.95	0.72	0.64
8	0.90	0.95	0.74	0.64
10	0.90	0.95	0.74	0.67
≥ 11	0.90	0.74	0.65	0.65

Tabella 3.14: *grid-search* per K-NN con ETAH come attributo mancante

λ	R^2 ETAD	R^2 ETAE	R^2 ETAG	R^2 ETAR
0.00001	0.90	0.97	0.74	0.67
0.0001	0.90	0.97	0.74	0.67
0.001	0.91	0.98	0.74	0.67
0.01	0.90	0.97	0.74	0.67
0.1	0.88	0.96	0.72	0.65

Tabella 3.15: *grid-search* per Lasso con ETAH come attributo mancante

Si riporta nella Tabella 3.16 il risultato della *grid-search* per i quattro rendimenti.

Parametri	ETAD	ETAE	ETAG	ETAR
λ	0.001	0.001	0.001	0.001
K	8	5	8	10

Tabella 3.16: Risultato della *grid-search* nel modello completo con l'attributo di analisi *ETAH* mancante

Dalle *grid-search* ottenute si hanno valori di R^2 più grandi quando le variabili mancanti sono ETAE o ETAD. Una volta decisi i valori di λ e K, si possono confrontare alcuni modelli di regressione, presentati nella Tabella 3.17 e scegliere quello più performante per ognuno dei rendimenti. Siccome ETAH dipende in modo cubico dai tre rendimenti presenti, si applica un algoritmo di regressione polinomiale utilizzando un polinomio di grado 3. Tuttavia, il modello generato è complesso perché vengono

svolte tutte le combinazioni possibili fra le variabili di ingresso al fine di ottenere dei polinomi di grado 3, per questo motivo viene applicato Lasso a seguito di questo algoritmo, in grado di mettere alcune variabili a zero per semplificare il modello.

Modello	R^2 ETAD	R^2 ETAE	R^2 ETAG	R^2 ETAR
Lineare	0.91	0.97	0.73	0.67
Lasso	0.91	0.97	0.72	0.65
Polinomiale + Lasso	0.91	0.97	0.74	0.68
K-NN	0.90	0.95	0.74	0.67

Tabella 3.17: Comparazione tra modelli di regressione per ETAH mancante

Il *framework* utilizza la tecnica Lasso quando il dato in ingresso non presenta ETAD o ETAE. Se invece manca ETAR, si utilizza una metodologia polinomiale affiancata da Lasso. Quando manca ETAG, non c'è alcuna differenza tra l'utilizzare Lasso con il modello polinomiare, oppure il K-NN.

Secondo caso: fattore forma mancante

Il secondo caso è rappresentato dall'arrivo di un nuovo certificato energetico che presenta fattore forma mancante. In questo caso i modelli di regressione presentano come input tutte le sette variabili utilizzate per il *Clustering*, meno il fattore forma da predirre. Il modello realizzato ha una forma di questo tipo:

$$S/V \sim UW + UO + SR + SD + ETAH + ANNO\,.$$

Vengono considerati dal *framework* i metodi di regressione lineare, Lasso e K-NN. Il primo step è sempre quello di attuare una *grid-search* per la scelta dei parametri degli algoritmi di regressione K-NN e Lasso. La Tabella 3.18 descrive il *tuning* dei parametri K per il K-NN e λ per la regressione Lasso.

λ	R^2 Lasso	K	R^2 K-NN
0.00001	0.75	1	0.78
0.0001	0.75	3	0.84
0.001	0.76	5	0.85
0.01	0.75	8	0.85
0.1	0.70	≥ 10	0.86

Tabella 3.18: *grid-search* per la regressione Lasso e K-NN con il fattore forma mancante

In assenza della variabile fattore forma, si può dedurre dalla tabella come i parametri migliori siano $\lambda = 0.001$ e K pari a 10. La tabella seguente mostra i diversi valori di R^2 per i modelli lineare, Lasso e K-NN. Si può evincere come il K-NN dia i risultati migliori, ed è quindi quello scelto dal *framework* per l'implementazione.

Modello	Valore R^2
Lineare	0.74
Lasso	0.76
K-NN	0.86

Tabella 3.19: Confronto tra gli algoritmi di regressione lineare, Lasso e K-NN con l'attributo di analisi fattore forma mancante

Terzo caso: altri attributi mancanti

Sono stati analizzati dei metodi di regressione per predirre altri attributi mancanti, come le trasmittante opache e trasparenti, che si sono rivelati poco soddisfacenti nei risultati ottenuti. Il *framework* in definitiva non li implementa per evitare di inserire del rumore all'interno del *dataset*.

3.4.3 Certificati Recuperati

I modelli di regressione e classificazione possono essere utilizzati non solo per caratterizzare attestati energetici non presenti nel *dataset*, ma anche per recuperare quei certificati che sono stati esclusi durante la fase di *preprocessing*. Infatti se un attributo è fuori dai range di ammissibilità è come se esso avesse un valore mancante che sarebbe possibile predirre.

Sigla	**Attributo**	**Eliminati (%)**
S/V	Fattore Forma	0.18
UW	Trasmittanza Trasparente	0.22
UO	Trasmittanza Opaca	0.15
ETAD	Rendimento Distribuzione	0.88
ETAE	Rendimento Emissione	0.72
ETAR	Rendimento Regolazione	0.20
ETAG	Rendimento Generazione	6.31

È possibile recuperare le certificazioni energetiche escluse per avere uno dei quattro rendimenti, oppure il fattore forma, fuori dagli intervalli di ammissibilità. Le

performance migliori sono ottenute, per il primo caso, quando mancano ETAE o ETAD.

Sigla	**Attributo Mancante**	**Variabile Target**	**Recuperati (%)**
S/V	Rapporto S/V	S/V	0.18
ETAD	Rendimento Distribuzione	ETAH	0.84
ETAE	Rendimento Emissione	ETAH	0.67

Tabella 3.20: Numero di certificati energetici recuperati con i modelli di regressione

Sono stati recuperati circa l'1,69% dei certificati energetici, a partire dall'1,78%. Questo è dovuto alla presenza di valori *outlier* presenti in altre variabili di interesse. L'ultimo step del modello completo è quello della classificazione mediante etichetta di *Clustering* attraverso il metodo visto in precedenza.

Capitolo 4

Conclusioni e sviluppi futuri

L'obiettivo di questo lavoro di tesi è stato quello di progettare e sviluppare una metodologia per la caratterizzazione energetica degli edifici localizzati nella città di Torino, in grado di supportare l'analista per analizzare un'elevata mole di dati relativi a certificati energetici. Si è sviluppato il *framework* TUCANA (TUrin Certificates ANAlysis), un *tool* automatico che attraverso diversi blocchi permette di analizzare l'efficienza energetica degli edifici residenziali della città di Torino, tramite tecniche di *Data Mining*. Al fine di ottenere un *dataset* affidabile, è stata effettuata una fase di pulizia di dati proveniente da fonti *open*. La fase di *data analytics* ha consentito di estrarre della conoscenza nascosta e di mostrarla ai vari utilizzatori, attraverso la navigazione di mappe interattive. Tale fase, grazie al processo di *Clustering* suddivide gli edifici in gruppi aventi caratteristiche termo-fisiche simili. Grazie alla rappresentazione dei dati su mappe geolocalizzate, l'analista riceve un valido supporto al *decision-making*, potendo rilevare gli aspetti più significativi dei dati. L'applicazione web sviluppata permette di rendere fruibile la conoscenza nascosta all'interno dei dati attraverso mappe navigabili interattive e altri strumenti statistici, a diversi *stakeholder*. Tale conoscenza può essere sfruttata in modo diverso da diversi utilizzatori, in basse alle loro esigenze. Ad esempio, i cittadini potrebbero essere interessati alle analisi energetiche di edifici localizzati in determinate aree della città, al fine di attuare delle scelte mirate su dove comprare o affittare casa. Gli analisti energetici potrebbero utilizzare il *framework* TUCANA per caratterizzare, sia con strategie *supervised*, che con tecniche *unsupervised*, gruppi di edifici con caratteristiche simili, in modo da realizzare analisi comparative. Infine, la pubblica amministrazione potrebbe volere effettuare degli interventi migliorativi in zone scarsamente performanti. Il *framework* TUCANA include anche un blocco di

Machine Learning per generalizzare la conoscenza estratta attraverso ragionamenti induttivi, al fine di supportare l'analista nel classificare nuovi edifici in modo preciso, in seguito all'aver realizzato dell'esperienza su un insieme di dati di *training* relativi alla città di Torino.
Possibili sviluppi futuri del *framework* riguardano:

- Nuovi modelli di generalizzazione per aumentare la precisione dei modelli proposti.
- Applicare la metodologia proposta anche ad edifici di altre destinazioni d'uso, e caratterizzare non solo la città di Torino ma l'intera regione Piemonte. In aggiunta, sarebbe interessante integrare i dati del catasto con quelli degli impianti, così da valutare altre discriminanti per valutare l'efficienza energetica.
- Prendere in considerazione altri algoritmi di *Clustering*, come quelli gerarchici, potrebbe consentire un'ulteriore caratterizzazione degli edifici.
- Estendere la metodologia anche ad altre città, per aumentare la conoscenza estratta e confrontare le performance delle diverse zone d'Italia.
- Integrare la metodologia proposta ad altri dati open (come l'inquinamento, le condizioni meteorologiche, la mobilità intelligente...) nell'applicazione web sviluppata.

Bibliografia

[1] *Energy performance of buildings*, URL:https://ec.europa.eu/energy/en/topics/energy-efficiency/energy-performance-of-buildings

[2] *Certificazione energetica (parte 1): quadro normativo e glossario*, URL: http://biblus.acca.it/focus/attestato-prestazione-energetica-ape/

[3] *Bollettino Ufficiale Regione Piemonte n. 22 del 31/05/2007*

[4] *Certificazione energetica, il Piemonte si adegua alle nuove norme*, URL: http://biblus.acca.it/certificazione-energetica-piemonte/

[5] *Prestazione energetica degli edifici residenziali*, URL: https://www.certificato-energetico.it/articoli/prestazione-energetica.html

[6] *Gradi Giorno e Zone Climatiche*, URL: https://www.posaqualificata.it/gradi-giorno-e-zone-climatiche/

[7] *Efficienza energetica degli edifici Teoria e Legislazione*, G.F.T.A. GruppoFisicaTecnicaAmbientale, Tecnograph Editore

[8] *Decreto del 26 giugno 2015, Articolo 3, Allegato 1*, URL: https://www.gazzettaufficiale.it/eli/id/2015/07/15/15A05200/sg

[9] *Usama Fayyad, Gregory I. Piatetsky-Shapiro e Padhraic Smyth. «Knowledge Discovery and Data Mining: Towards a Unifying Framework». In: (1996), pp. 8288*, URL: http: www.aaai.org/Papers/KDD/1996/KDD96-014.pdf.

[10] *Vladimir I Levenshtein. «Binary codes capable of correcting deletions, insertions, and reversal». In: Soviet Physics-Doklady (1966), pp. 707710*

[11] *Charu C. Aggarwal «Outlier Analysis». In: Data Mining (2015), pp. 237-263*

[12] *Bernard Rosner. «Percentage points for a generalized ESD many-outlier procedure». In: Technometrics 25 (1983), pp. 165172*

[13] *Sheldon M. Ross. Probabilità e statistica per lingegneria e le scienze. Maggioli Editore, 2015*

[14] *Martin Ester Hans-Peter Kriegel Jiirg Sander Xiaowei X. «A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise». In: KDD-96 Proceedings (1996)*

[15] *Jacek Biesiada e Wlodzislaw Duch. «Feature Selection for High-Dimensional Data A Pearson Redundancy Based Filter». In: Computer Recognition Systems 2 (2007), pp. 242-249*

[16] *Luai Shalabi, Shaaban Zyad e Basil Al-Kasasbeh. «Data Mining: A Preprocessing Engine». In: Journal of Computer Science 2 (set. 2006)*

[17] *Tan, Steinbach, Kumar. «Introduction to Data Mining». McGraw Hill 2006*

[18] Purnima Bholowalia e Arvind Kumar. «EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN». In: International Journal of Computer Applications 105 (2014), pp. 1724

[19] Satopaa, Ville, et al. "Finding a" kneedle" in a haystack: Detecting knee points in system behavior." 2011 31st International Conference on Distributed Computing Systems Workshops. IEEE, 2011.

[20] *Leo Breiman. Classification and Regression Trees. Routledge, 2017*

[21] *Tukey J.W. Hoaglin D. Mosteller F. «Understanding robust and exploratory data analysis». In: (1983)*

[22] *Leon Shklar, Richard Rosen «Web Application Architecture: Principles, Protocols and Practices». John Wiley and Sons Ltd 2009*

[23] *Miguel Grinberg «Flask Web Development: developing web applications with Python». O'Reilly 2018*

[24] *Douglas C. Montgomery, Elizabeth A. Peck and G. Geoffrey Vining,Introduction to Linear Regression Analysis, Wiley 2012*

[25] *Alfio Quarteroni, Matematica Numerica, Esercizi, Laboratori e Progetti, Seconda edizione, Springer 2013, pp. 199-252*

[26] *Hoerl, Arthur E., and Robert W. Kennard. "Ridge regression: Biased estimation for nonorthogonal problems." Technometrics 12.1 (1970): 55-67.*

[27] *Santosa, Fadil, and William W. Symes. "Linear inversion of band-limited reflection seismograms." SIAM Journal on Scientific and Statistical Computing 7.4 (1986): 1307-1330.*

[28] *Tibshirani, Robert. "Regression shrinkage and selection via the lasso." Journal of the Royal Statistical Society: Series B (Methodological) 58.1 (1996): 267-288*

[29] *Rui Li and Guan Gong, K-Nearest-Neighbour Non-Parametric Estimation of Regression Functions in the Presence of Irrelevant Variables, The Econometrics Journal (2008), pp. 396-408.*

[30] *Draper, Norman R. and Harry Smith. Applied regression analysis. Vol. 326. John Wiley Sons, 1998.*

[31] Leif E. Peterson, K-Nearest Neighbor, Scholarpedia 2009.

[32] McKinney, Wes. "pandas: a Python data analysis library." see http://pandas. pydata. org (2015)

[33] Tosi, Sandro. Matplotlib for Python developers. Packt Publishing Ltd, 2009

[34] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." Journal of machine learning research 12.Oct (2011): 2825-2830

[35] Filipe and M. J. et al. python-visualization/folium: v0.6.0, Aug. 2018.

[36] Ihaka, Ross, and Robert Gentleman. "R: a language for data analysis and graphics." Journal of computational and graphical statistics 5.3 (1996): 299-314.

[37] Kotu, Vijay, and Bala Deshpande. Predictive analytics and data mining: concepts and practice with rapidminer. Morgan Kaufmann, 2014.

[38] *Fattori di conversione in energia primaria dellenergia termica fornita ai punti di consegna della rete di teleriscaldamento della rete di Torino. 2018. URL: https://www.gruppoiren.it/documents/21402/69847/PEF+2018+-+Torino.pdf/4bc7fbb3-2748-4319-bfa3-44530f3559ba*