POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Gestionale

Tesi di Laurea Magistrale

Esplorazione di una base di dati delle Certificazioni energetiche mediante tecniche di analisi dei dati

Relatori
Chiar.ma Prof.ssa Tania Cerquitelli
Dott.ssa Evelina Di Corso

Candidato
Mirko Deleuchi

Luglio 2018

Ai miei genitori.
Miei mentori, miei eroi.

Ringraziamenti

Vorrei spendere due parole per tutte le persone che mi hanno accompagnato in questo percorso di laurea magistrale.
Desidero esprimere i miei più sinceri ringraziamenti alla Prof.ssa Tania Cerquitelli per avermi dato l'opportunità di cimentarmi nella presente tesi. Un'esperienza che mi ha fatto crescere molto sotto il profilo professionale.
Ringrazio la Dott.ssa Evelina Di Corso, che con pazienza e gentilezza mi ha dato preziosissimi consigli e dalla quale ho imparato molto in questi mesi.
Ringrazio la mia famiglia: i miei genitori, mia sorella, le mie nonne e mia zia, che hanno sempre creduto in me e in questi anni non mi hanno fatto mai mancare nulla, dandomi la possibilità di realizzarmi con serenità e senza alcun tipo di pressione.
Vorrei ricordare i colleghi che mi hanno accompagnato in questo percorso e che hanno reso meno pesanti le ore dedicate allo studio: Simone, Livio, Giorgia, Camilla B., Camilla F. ed Arianna.
Ringrazio i miei amici, che attraverso il nostro gruppo WhatsApp non mi fanno pesare la distanza.
Vorrei ringraziare Federica, che da coinquilina è diventata un'ottima amica.
Infine, ma non per minor importanza, ringrazio la mia ragazza Giusy, che non mi ha mai lasciato solo nei momenti di maggiore stress, riuscendo a trasmettermi calma e supportandomi nelle gionate di sconforto.

Indice

Elenco delle tabelle

Elenco delle figure

Introduzione

Le *Certificazioni energetiche* riferite agli edifici sono delle procedure che consentono di effettuare una stima della performance energetica di un palazzo a partire dalle sue geometrie e dalle sue proprietà termiche e fisiche. La certificazione energetica nasce per migliorare l'efficienza energetica nell'edilizia, visto che nel nostro continente essa consuma circa il 40% dell'energia prodotta. Le normative europee forniscono agli stati membri le linee guida, è poi compito di questi ultimi legiferare in tale campo a seconda delle peculiarità del proprio territorio. Le leggi nazionali, a loro volta ,vengono tradotte ed adattate dalle Regioni che definiscono i processi con cui viene prodotto l'*Attestato di Prestazione Energetica (A.P.E.)*, documento che certifica l'efficienza energetica dell'edificio.

Grazie all'attuazione della normativa europea, è cresciuta la disponibilità di dati riguardanti le caratteristiche edilizie ed impiantistiche. I dati vengono raccolti in database, ciò ha aperto la strada a numerosi studi sul miglioramento dell'efficienza energetica degli edifici. In questo ambito, gli studi effettuati nella presente tesi si pongono come obiettivo quello di automatizzare e standardizzare l'emissione degli APE, individuando le caratteristiche fisiche ed energetiche degli edifici che determinano la performance energetica. Attualmente, gli APE vengono rilasciati dai certificatori energetici, i quali dopo aver raccolto le informazioni relative agli edifici rilasciano nell'APE una valutazione di sintesi detta *Classe Energetica.* Dimostreremo che questa procedura presenta delle lacune ed in alcuni casi porta il certificatore a ricadere in alcuni errori. Attraverso tecniche statistiche quali la *Regressione Multipla* ed algoritmi di clustering come il *K-Means*, mostreremo quali sono le caratteristiche termo-fisiche e geometriche responsabili della performance energetica; con l'ausilio di uno strumento di *cross-validation*, vedremo con quali regole vengono formati i cluster ed aiutandoci con differenti strumenti grafici visualizzeremo come sono caratterizzati i singoli cluster.

Per raggiungere tali obiettivi, spiegheremo come è stato realizzato il framework *F-SCAN* in modo da creare un processo automatico con cui rilasciare gli APE.

La presente tesi è strutturata in 5 capitoli.

Nel **Capitolo 1** viene introdotta la certificazione energetica in ambito nazionale e viene focalizzata la normativa vigente in Piemonte nel 2013, anno in cui sono stati rilasciati i certificati da cui abbiamo estratto i dati.

Nel **Capitolo 2** vengono richiamati i concetti teorici del processo di estrazione della conoscenza, nel quale riveste un'importanza particolare l'applicazione degli algoritmi di Data Mining.
Nel **Capitolo 3** vengono presentati brevemente i software che sono stati utilizzati per l'implementazione del framework.
Nel **Capitolo 4** viene presentato il framework *F-SCAN*, un'architettura che utilizza in primo luogo tecniche statistiche ed algoritmi di clustering basati sulla densità per ottimizzare la selezione dei dati nel database; in secondo luogo algoritmi di clustering partizionativi ed alberi di decisione per caratterizzare gli edifici della Regione Piemonte.
Il **Capitolo 5** analizza nel dettaglio i risultati di due esperimenti scelti fra i tanti che sono stati effettuati tramite l'utilizzo dell'*F-SCAN* su due dataset reali.
Il **Capitolo 6**, infine, presenta le conclusioni dell'elaborato e gli sviluppi futuri.

Capitolo 1

La Certificazione energetica degli edifici

La certificazione energetica degli edifici è una procedura di valutazione che ha lo scopo di stimolare il miglioramento dell'efficienza, grazie alle informazioni sui consumi energetici. Queste vengono fornite ai proprietari degli edifici ed agli utilizzatori, con l'obiettivo di raggiungere e mantenere condizioni ambientali ottimali interne agli edifici stessi. Tale certificazione è parte di una serie di iniziative volte alla tutela dell'ambiente: oggi, infatti, il tema della salvaguardia del nostro pianeta è al centro dei dibattiti politici internazionali. A partire dal 1997, con il Protocollo di Kyoto, la quasi totalità delle nazioni si impegna ad attuare una serie di normative volte alla riduzione dell'impatto delle attività umane sull'incremento della temperatura globale. I telegiornali, le radio, le pubblicità istituzionali cercano di sensibilizzare gli individui riguardo al tema, ma ciò non basta: servono iniziative come la certificazione energetica, che obbliga proprietari ed utenti ad attenersi alla normativa.

La particolare attenzione prestata agli edifici è giustificata dal fatto che il 40% dei consumi finali globali energetici dell'Unione Europea è impiegato principalmente per essi. La certificazione nasce con la *direttiva comunitaria 2002/91/CE* relativa al rendimento energetico nell'edilizia, che aveva lo scopo di allineare la normativa nazionale degli stati membri dell'UE con la normativa europea. Il decreto legislativo con il quale è stata avviata tale direttiva è entrato in vigore nel 2005 ed ha subìto modifiche successive. Lo scopo iniziale era quello di stabilire i criteri, le condizioni e le modalità per incentivare il mercato immobiliare a migliorare il rendimento energetico degli edifici al fine di: favorire lo sviluppo e l'integrazione delle fonti rinnovabili; contribuire a conseguire gli obiettivi nazionali di limitare i gas serra, come previsto dal protocollo di Kyoto; promuovere lo sviluppo tecnologico di nuove fonti di energia. I dati utili a comporre le certificazioni energetiche sono raccolti e gestiti in un catasto. Dal 2005 in avanti, si è verificata una crescita continua di tali dati e sono stati fatti moltissimi studi ed esperimenti su di essi.

In Italia, la legislazione ha consentito alle Regioni e alle Province autonome di utilizzare le proprie competenze per calare la normativa europea nella realtà locale. Infatti, l'Italia è un paese molto diversificato da un punto di vista climatico e questo comporta una grande difficoltà nella definizione di standard a livello nazionale. Delegare le Regioni e le Province autonome, ha come vantaggio principale quello di definire gli standard appropriati al particolare clima regionale; d'altro canto lo svantaggio è quello di non poter confrontare certificazioni energetiche appartenenti a regioni italiane che si trovano in differenti fasce climatiche: è verosimile pensare che un edificio ubicato in Sicilia abbia bisogno di meno energia rispetto ad un edificio ubicato in Val d'Aosta, ma questo potrebbe dipendere solamente dal fatto che la Sicilia ha una temperatura media annua superiore di 14°C rispetto alla Val d'Aosta e non perché l'edificio siciliano è più performante di quello valdostano. In seguito, vedremo che tramite l'utilizzo di alcune misure, è possibile fare un confronto fra edifici ubicati in regioni diverse, tuttavia resta il fatto che tali misure costituiscono delle approssimazioni.
La certificazione, in termini pratici, sintetizza il fabbisogno annuo di energia necessaria per soddisfare i servizi di climatizzazione invernale ed estiva, il riscaldamento dell'acqua per uso domestico, la ventilazione e l'illuminazione. Il fabbisogno energetico di un edificio dipende dalla sua dotazione impiantistica, dai materiali edilizi utilizzati per la sua realizzazione che ne determinano l'isolamento termico, dalla sua posizione e dalle caratteristiche di illuminazione. La normativa suggerisce anche come migliorare il rendimento.
Per ottenere la certificazione energetica occorre rivolgersi al certificatore energetico. Questo invierà un tecnico (spesso un geometra) che tramite un sopralluogo rileva e registra i parametri necessari per il calcolo della certificazione. Nel caso in cui l'edificio è ancora in fase di progettazione, si effettua la procedura di calcolo da progetto: i dati vengono reperiti dal progetto energetico (relazione chiamata *legge 10*). La classificazione energetica riassume il rendimento energetico dell'edificio tramite la classe energetica. Essa è una lettera che va da A+ a G. Dal 2015, la classificazione dipenderà da quanto l'immobile è più o meno performante rispetto ad un edificio di riferimento con caratteristiche medie.

1.1 La Certificazione energetica in Piemonte

Prendiamo in considerazione la regione Piemonte, perché il nostro dataset è stato estratto proprio dai dati catastali piemontesi. In Piemonte, la certificazione energetica è disciplinata dalla Legge 28 maggio 2007, n.13, *Disposizioni in materia di rendimento energetico nell'edilizia*, con la quale si individuano gli indirizzi, le disposizioni e gli strumenti necessari al miglioramento delle performance energetiche degli edifici esistenti e di quelli che sono in procinto di essere costruiti. Inoltre la certificazione energetica, con tale legge, è divenuta obbligatoria per gli edifici di

nuova costruzione, per gli edifici esistenti che devono subire una ristrutturazione edilizia e per tutti gli edifici soggetti a locazione o compravendita.
La Regione Piemonte ha lavorato a tale legge sposando il *Piano 20 20 20*: si tratta dell'insieme delle misure pensate dall'UE per raggiungere gli obiettivi fissati dal protocollo di Kyoto. Il Piemonte, pertanto, si impegna a ridurre le emissioni dei gas serra del 20%, alzare la quota di energie rinnovabili al 20% e portare al 20% il risparmio energetico entro il 2020. Per centrare questi obiettivi ambiziosi, il Settore Politiche energetiche guida i programmi di intervento regionale nel campo energetico, studia quali incentivi possono essere offerti per aumentare l'efficienza energetica incrementando l'utilizzo delle fonti rinnovabili e riducendo i consumi di CO_2, coordina lo sviluppo delle infrastrutture e delle reti energetiche nel territorio ed organizza campagne di sensibilizzazione sulle tematiche energetiche utilizzando differenti canali di divulgazione, in modo da influenzare il maggior numero di persone possibili.
Per attuare quanto detto, il settore regionale di competenza ha progettato ed implementato un Sistema informativo condiviso dove memorizzare i dati relativi alla Certificazione energetica. Questo sistema, chiamato Sistema per la Certificazione Energetica degli Edifici (*SICEE*), comprende il *Catasto energetico degli edifici della Regione Piemonte*, un database dove sono memorizzati i dati che fanno riferimento alle performance energetiche degli edifici esistenti e di quelli di nuova costruzione. I dati allocati nel Catasto, si riferiscono al fabbisogno energetico stimato durante il processo di certificazione ed al consumo reale di energia determinato dal certificatore, il quale effettua il calcolo sui dati reali.
La Regione Piemonte, tramite l'Agenzia Regionale per la Protezione Ambientale (A.R.P.A.), effettua delle verifiche a campione per accertare la regolarità dei certificati emessi e, nel caso in cui gli edifici siano di nuova costruzione, la conformità delle opere realizzate rispetto alle specifiche progettuali.

1.2 Descrizione dei dati di certificazione utilizzati

In una certificazione energetica, sono presenti grandi quantità di attributi. Infatti, all'interno di una certificazione, troviamo: dati catastali, dati tecnici generali, indici di fabbisogno ed informazioni sull'energia proveniente da fonti rinnovabili. Nella fase di selezione degli attributi (si veda il paragrafo 2.2) vengono individuate le caratteristiche termo-fisiche da includere nell'analisi dei dati, coerentemente con gli obiettivi che l'analisi stessa si pone. Precisiamo che tutti gli esperimenti sono stati effettuati utilizzando i dati provenienti da certificazioni energetiche riferite ad edifici adibiti esclusivamente ad uso residenziale. In tabella 1.1 riportiamo la classificazione generale degli edifici per categorie.
Descriviamo di seguito le informazioni presenti nei dataset su cui sono stati eseguiti gli esperimenti:

Classificazione generale degli edifici per categorie	
E.1	Edifici adibiti a residenza e assimilabili
E.2	Edifici adibiti a uffici e assimilabili
E.3	Edifici adibiti a ospedali, cliniche o case di cura e assimilabili
E.4	Edifici adibiti ad attività ricreative o di culto e assimilabili
E.5	Edifici adibiti ad attività commerciali e assimilabili
E.6	Edifici adibiti ad attività sportive e assimilabili
E.7	Edifici adibiti ad attività scolastiche a tutti i livelli e assimilabili
E.8	Edifici adibiti ad attività industriali e artigianali e assimilabili

Tabella 1.1: Classificazione degli edifici per destinazione d'uso secondo la normativa D.P.R. n.412/93.

- **Volume lordo riscaldato** *(V)*: si misura in metri cubi, rappresenta la volumetria lorda dell'edificio riscaldata;
- **Superficie disperdente totale** *(S)*: si misura in metri quadrati, è la superficie che separa gli ambienti interni dell'edificio dall'ambiente esterno e dagli ambienti non riscaldati. Il calore viene disperso attraverso di essa;
- **Superficie utile** *(S_u)*: si esprime in metri quadrati, rappresenta la superficie calpestabile riscaldata all'interno dell'edificio;
- **Altezza Media** *(V/S_u)*: è data dal rapporto tra il volume lordo riscaldato V e la superficie utile S_u, esprime la distanza che intercorre tra l'estradosso del pavimento e l'intradosso del soffitto;
- **Fattore Forma** *(S/V)*: è dato dal rapporto tra la superficie disperdente totale S ed il volume lordo riscaldato V, la sua unità di misura è il reciproco del metro (m^{-1}). Rappresenta la quantità di superficie che racchiude il volume lordo riscaldato. A seconda del proprio fattore forma, ogni edificio si pone in modo diverso rispetto al contesto climatico ed ambientale in cui è ubicato. Tale grandezza, in particolare, influenza significativamente le perdite termiche: quanto più è elevato il fattore forma, tanto più è elevato lo scambio di calore tra gli ambienti interni riscaldati e l'ambiente esterno. La forma ottimale per la conservazione del calore, ossia per minimizzare la dispersione energetica in relazione al coefficiente di forma, risulta essere la forma sferica; dopo di essa, sono le forme cubiche caratterizzate da un fattore forma basso ad essere preferibili. A parità di caratteristiche strutturali, il fattore forma decresce all'aumentare delle dimensioni: ciò è particolarmente intuitivo, basti pensare infatti che la superficie varia con il quadrato della sua dimensione, mentre il volume con il cubo. Prendendo in considerazione un edificio condominiale, si può dedurre che gli appartamenti che si trovano all'interno avranno

un fattore forma differente: quelli che avranno le proprie pareti a contatto con gli ambienti non riscaldati avranno un fattore forma maggiore rispetto agli appartamenti che sono a contatto con altri ambienti riscaldati. In sintesi, il fattore forma permette di paragonare soluzioni costruttive differenti dal punto di vista strutturale, al netto di tutte le altre caratteristiche (isolamento termico e condizioni atmosferiche);

Le prossime grandezze che analizzeremo sono le trasmittanze opache e le trasmittanze trasparenti. Prima di passare alla spiegazione di queste due caratteristiche, occorre precisare che cosa intendiamo per trasmittanze e a cosa serve il loro calcolo. Un concetto fondamentale nello studio della performance energetica è l'isolamento termico. Uno degli obiettivi principali di chi si occupa di efficienza energetica, è quello di contenere la dispersione di energia sfruttando le resistenze dei materiali. Nel nostro caso, facciamo ovviamente riferimento all'energia termica: sfruttando le resistenze termiche dei materiali, si vuole trattenere la maggiore quantità di calore possibile negli ambienti riscaldati. La *trasmittanza termica U* è una grandezza che calcola gli effetti generati dagli scambi d'aria tra la parete dell'ambiente riscaldato e la parete dell'ambiente non riscaldato. Si misura in W/m^2K, rappresenta cioè la quantità di energia termica dispersa in una superficie pari ad un metro quadro per una differenza di temperatura pari ad un grado Kelvin. Nel processo di certificazione energetica, il tecnico calcola la trasmittanza separatamente per le superfici opache e trasparenti, poiché esse dipendono da fattori differenti: le prime dipendono dal materiale utilizzato e dalla tecnica di costruzione; le seconde dal tipo di vetro utilizzato, dal telaio, e dall'utilizzo di distanziatori nei serramenti. Un edificio a trasmittanza bassa è più performante rispetto ad un edificio a trasmittanza alta. Nella certificazione troviamo i seguenti valori:

- **Trasmittanze opache** *(U_{op})*: indica la trasmittanza media ponderata delle superfici opache confinanti con l'ambiente esterni (o con ambienti non riscaldati);
- **Trasmittanze trasparenti** *(U_w)*: indica la trasmittanza media ponderata delle superfici e delle chiusure trasparenti confinanti con l'ambiente esterno (o con ambienti non riscaldati);

Descriviamo ora i rendimenti, gli indici di fabbisogno e gli indici di prestazione energetica che sono presenti nei dataset utilizzati:

- **Rendimento di generazione decimale** *(η_{gn})*: rappresenta il fenomeno della dispersione energetica a livello di generazione. L'energia che viene fornita al generatore, successivamente viene trasferita ad un fluido termovettore per il trasporto. Durante tale passaggio si verifica una perdita di calore causata da molteplici fenomeni, quali perdite al camino, perdite al bruciatore, ecc.

Maggiore è il rendimento di generazione, minore sono le perdite energetiche che si verificano tra il generatore ed il fluido;

- **Rendimento di distribuzione decimale** *(η_d)*: caratterizza le perdite che si verificano nella rete di distribuzione, che trasferiscono il fluido termovettore dal generatore ai dispositivi di emissione. In questa fase, il calore viene disperso dai tubi, spesso non isolati, verso ambienti non riscaldati o verso l'ambiente esterno;

- **Rendimento di regolazione decimale** *(η_e)*: tale grandezza viene calcolata come rapporto tra la quantità di calore richiesta per scaldare gli ambienti con una regolazione teorica perfetta e la quantità di calore richiesta per scaldare gli stessi ambienti con una regolazione reale. La termoregolazione modifica l'emissione del calore proveniente dal corpo scaldante nel momento in cui viene rilevata una variazione di calore proveniente da una fonte endogena rispetto all'impianto. La regolazione è perfetta quando la modulazione dell'emissione è istantanea, nella realtà tuttavia intercorre un determinato intervallo di tempo tra la ricezione della variazione di calore nell'ambiente e l'inizio dell'azione del dispositivo di termoregolazione. Un rendimento elevato indica un'elevata capacità del sistema di regolazione nel recepire variazioni del carico termico in un intervallo di tempo breve;

- **Rendimento di emissione decimale** *(η_{rg})*: è dato dal rapporto tra la quantità di calore richiesta per riscaldare gli ambienti con un sistema di emissione teorico e la quantità di calore richiesta per riscaldare lo stesso ambiente con un sistema di emissione reale. Il rendimento di emissione definisce la modalità con cui lo scambio di calore tra il dispositivo di erogazione e l'ambiente interno aumenta la quantità di energia termica che il terminale deve fornire in confronto a quella teorica richiesta. Nel sistema di emissione, infatti, si possono generale delle perdite che dipendono dal tipo di terminale e dalle condizioni di funzionamento;

- **Rendimento globale riscaldamento Torino** *($\eta_{g,To}$)*[1] : è calcolato come rapporto tra il fabbisogno di calore utile per il riscaldamento invernale ed

[1] I dati caratterizzati dal pedice "To" fanno riferimento ad una localizzazione teorica dell'immobile. Con l'obiettivo di rendere confrontabili i dati appartenenti a certificazioni energetiche effettuate per edifici ubicati in zone differenti , si esegue un'approssimazione utilizzando i gradi giorno. Il grado giorno (GG) di una località, è la somma estesa a tutti i giorni, in un periodo annuale convenzionale di riscaldamento, delle sole differenze positive giornaliere tra la temperatura, fissata convenzionalmente per ogni paese, e la temperatura media esterna giornaliera. Ogni volta che incontreremo un indice che presenta nel nome "Torino" o nel pedice del suo simbolo "To", significa che i dati utilizzati per il suo calcolo sono stati moltiplicati per i GG di Torino e divisi per i GG del comune di appartenenza.

il corrispondente fabbisogno di energia primaria durante la stagione di riscaldamento, compresa l'energia elettrica degli apparati ausiliari. É dato dal prodotto dei quattro rendimenti precedenti: $\eta_{g,To} = \eta_{gn,To} \times \eta_{d,To} \times \eta_{e,To} \times \eta_{rg,To}$.

- **Rendimento globale acqua calda sanitaria** *($\eta_{g,W}$)*: è il rendimento dell'impianto che fornisce acqua calda, è espresso in formato decimale come i rendimenti che abbiamo introdotto in precedenza;

- **Rendimento globale riscaldamento ed acqua calda sanitaria** *($\eta_{g,R,W}$)*: è il rendimento globale dell'impianto di produzione dell'acqua calda sanitaria e dell'impianto di climatizzazione. Anche in questo caso, esso è espresso in formato decimale e non come percentuale;

- **Rendimento stagionale acqua calda sanitaria Torino** *($\eta_{s,W,TO}$)*: è il rendimento medio stagionale dell'impianto di produzione dell'acqua calda sanitaria, corretto con i GG al fine di rendere confrontabili edifici ubicati in zone climatiche differenti. Espresso in termini decimali;

- **Rendimento medio globale stagionale acqua calda sanitaria** *($\eta_{g,s,W}$)*: è il rendimento medio stagionale dell'impianto di produzione dell'acqua calda sanitaria. Espresso come valore decimale;

- **Fabbisogno energia termica utile** *(Q_h)*: è la quantità globale di energia termica annua necessaria affinché gli ambienti interni mantengano una temperatura pari alla temperatura di progetto, considerando un regime di funzionamento continuo. Questa grandezza è da considerare ideale: in primo luogo perché l'impianto di riscaldamento non è continuamente in funzione; in secondo luogo perché tale indice fa riferimento a condizioni di temperatura uniformi nell'ambiente considerato. Viene misurato in kWh/m^2 o in kWh/m^3;

- **Fabbisogno energia termica utile acqua calda sanitaria** *($Q_{h,W}$)*: è la quantità globale di energia termica annua necessaria per l'erogazione di acqua calda, utilizzata per scopi igienico-sanitari. Viene misurato in kWh;

- **Fabbisogno energia termica utile Torino** *($Q_{h,To}$)*: è il fabbisogno di energia termica utile moltiplicato per i GG di Torino e diviso per i GG del comune di appartenenza;

- **Fabbisogno acqua calda sanitaria soddisfatto da fonti rinnovabili** *($Q_{W,FR}$)*:è la quota di energia termica, proveniente da fonti energetiche rinnovabili, che viene utilizzata per riscaldare l'acqua calda sanitaria;

- **Indice di prestazione energetica acqua calda sanitaria Torino** *($EP_{i,W,To}$)*: è il rapporto tra il fabbisogno annuale per la produzione di acqua calda sanitaria ($Q_{h,W}$), al netto della quota energetica prodotta dalle fonti di energia rinnovabili ($Q_{W,FR}$), e la superficie utile dell'edificio (S_u) o il suo volume lordo riscaldato (V), rapportato al valore del rendimento medio stagionale dell'impianto di produzione dell'acqua calda sanitaria. Espresso in kWh/m^2 o in kWh/m^3;

- **Indice di prestazione del riscaldamento Torino** *($EP_{i,To}$)*: espresso in kWh/m^2 o in kWh/m^3, è il rapporto tra il fabbisogno annuale per la produzione di energia termica utile Q_h e la superficie utile dell'edificio (S_u) o al suo volume lordo riscaldato (V),rapportato al valore del rendimento dell'impianto che produce energia termica. Tale indice è corretto con i GG per rendere confrontabili gli edifici che appartengono a città differenti;

- **Indice di prestazione energetica globale Torino***($EP_{L,To}$)*:è determinato automaticamente dal *SICEE*, è la somma tra l'indice di prestazione energetica del riscaldamento Torino ($EP_{i,To}$) e l'indice di prestazione energetica per l'acqua calda sanitaria Torino ($EP_{W,To}$), quest'ultimo non compare in lista perché non è incluso nel nostro dataset. É espresso in kWh annui;

- **Indice di prestazione energetica acqua calda sanitaria fonti rinnovabili Torino** *($EP_{i,W,To,FR}$)*: misura la quota di energia termica ottenuta da fonti energetiche rinnovabili per la produzione dell'acqua calda sanitaria. Espresso in kWh annui, è corretto con i GG;

- **Prestazione energetica acqua calda sanitaria check** *($EP_{i,W,check}$)*: è il valore accertato della prestazione energetica dell'acqua calda sanitaria, ovvero della quantità di energia termica prodotta dall'impianto di riscaldamento, utilizzata per riscaldare l'acqua calda sanitaria. Indice espresso in kWh

- **Potenza riscaldamento** *(W)*: indica la potenza termica erogata dagli impianti di riscaldamento per climatizzare gli ambienti interni. Si misura in watt (W);

- **Prestazione raggiungibile** EP_L^*: indica la quantità di energia termica erogabile teoricamente, qualora gli impianti di riscaldamento lavorassero in condizioni di efficienza;

- **Classe energetica**: è definita a partire dall'indice di prestazione energetica con localizzazione a Torino ($EP_{L,To}$). Tale indice, come già detto in precedenza, viene calcolato automaticamente dal *SICEE* e confrontato con i valori limite definiti nella tabella sottostante, la classe energetica viene assegnata di conseguenza:

Classe	**Valori limite**				
A+		$\leq$	$EP_{L,To}$	$<$	27 kWh/m^2
A	27 kWh/m^2	$\leq$	$EP_{L,To}$	$<$	44 kWh/m^2
B	44 kWh/m^2	$\leq$	$EP_{L,To}$	$<$	82 kWh/m^2
C	82 kWh/m^2	$\leq$	$EP_{L,To}$	$<$	143 kWh/m^2
D	143 kWh/m^2	$\leq$	$EP_{L,To}$	$<$	201 kWh/m^2
E	201 kWh/m^2	$\leq$	$EP_{L,To}$	$<$	249 kWh/m^2
F	249 kWh/m^2	$\leq$	$EP_{L,To}$	$<$	300 kWh/m^2
G	300 kWh/m^2	$\leq$	$EP_{L,To}$	$<$	436 kWh/m^2

Tabella 1.2: Valori di $EP_{L,To}$ per individuare la classe energetica secondo quanto riportato dalla normativa vigente

Capitolo 2

L'estrazione della conoscenza

2.1 Data Mining e Knowledge Discovery in Databases

Il termine *Data Mining* fa riferimento ad un processo complesso di estrazione della conoscenza. Al giorno d'oggi, la mole di dati prodotta sta diventando sempre più consistente. Il 90% dei dati esistenti sono stati prodotti negli ultimi due anni; se inoltre consideriamo che ogni minuto, in rete, vengono scambiate 200 milioni di email, visualizzati 60 ore di contenuti video su YouTube, fatte 9 milioni di telefonate e condivisi 300mila tweet[1], è addirittura difficile immaginare la quantità di dati che abbiamo a disposizione. Le capacità limitate dell'uomo rendono impossibile estrarre informazioni dai *Big Data*, per questo motivo si ricorre alle tecniche di Data Mining, con l'obiettivo di individuare le relazioni meno evidenti fra i dati rendendole esplicite.
Il Data Mining non è semplicemente un'analisi statistica di dati, ma costituisce una serie di tecniche complesse, automatiche o semiautomatiche, necessarie per l'estrazione delle informazioni con l'obiettivo di utilizzarle nel *decision making*. Tale insieme di processi, identifica trend e pattern. Un pattern è una rappresentazione sintetica e significativa di un insieme di dati. In genere, un pattern rappresenta un modello che ricorre spesso nei dati, ma con tale termine facciamo riferimento anche ad un modello eccezionale. Un pattern deve essere:

- valido: deve essere applicabile anche sui nuovi dati con un determinato grado di incertezza;
- nuovo: deve apportare una variazione nella conoscenza estratta;
- utile: l'utente deve essere in grado di prendere delle azioni di conseguenza;

[1] dati forniti dall'azienda Beantech all'inizio del 2015

- comprensibile: l'utente deve essere in grado di interpretarlo.

Spesso, il termine Data Mining viene considerato intercambiabile con il termine *Knowledge Discovery in Databases (KDD)*, ma ciò è un errore. Infatti, il Data Mining è solamente una fase del KDD e consente di applicare uno specifico algoritmo con l'obiettivo di identificare i pattern. Il risultato del processo di Data Mining è una generalizzazione concettuale dei dati. I dati vengono raggruppati in insiemi omogenei, presentanti caratteristiche comuni. Con *Knowledge Discovery in Databases (KDD)* intendiamo l'intero processo che consente di trattare i dati per poter estrarre la conoscenza. La figura 2.1 mostra sinteticamente lo schema del processo KDD.

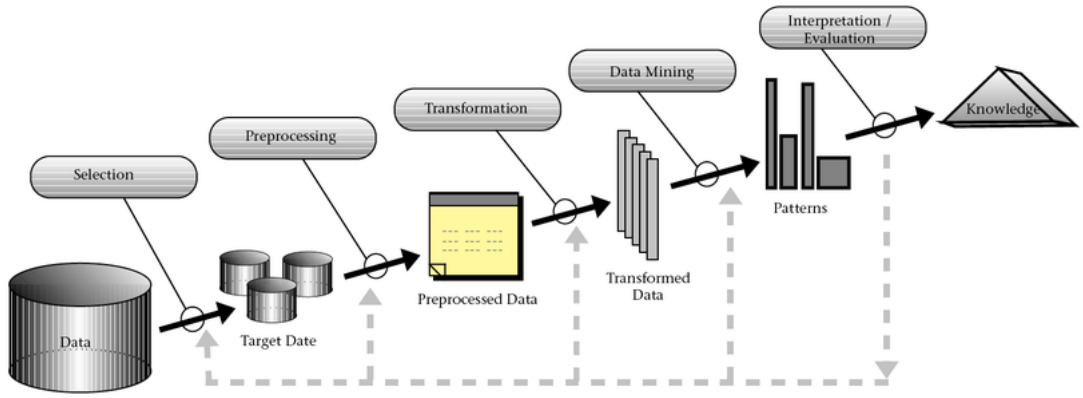


Figura 2.1: Il processo di Knowledge Discovery in Databases. ©Kumar

Il punto di partenza del KDD è la selezione dei dati rilevanti, questa viene effettuata a seconda dello scopo preposto all'analisi. Questa fase viene chiamata *selezione*. Le fasi successive prendono il nome di *preprocessing* e *trasformazione* dei dati: la prima serve per eliminare dai dati le informazioni che vengono ritenute inutili; la seconda invece serve per effettuare le opportune correzioni in modo che i dati estratti da database differenti non presentino inconsistenze informatiche. Durante la fase di trasformazione, possono essere incluse informazioni ulteriori, in modo che i dati siano navigabili. Solamente dopo queste tre fasi possiamo cominciare il processo di Data Mining, attraverso l'utilizzo di algoritmi per l'estrazione dei pattern significativi che, in fase successiva, devono essere soggetti all'interpretazione ed alla validazione da parte dell'utente: solo in questo modo le informazioni estratte diventano conoscenza, utilizzabile per il decision making.
Durante le fasi preliminari di elaborazione e di trasformazione si utilizzano tecniche e strumenti che prendono il nome di processi di ETL (*Extract, Transform, Load*).

2.2 Selezione, preprocessing e trasformazione

Una volta che sono stati individuati gli ambiti di applicazione del KDD, occorre fissare gli obiettivi in virtù dei quali si progetta il processo di estrazione della conoscenza. É una fase tutt'altro che semplice: al contrario di quello che si può pensare, l'identificazione degli obiettivi richiede sforzi considerevoli in termini di risorse e di tempo. Identificati gli obiettivi, si procede con la selezione dei dati che devono essere soggetti all'analisi, per effettuare su di essi le operazioni di preprocessing e di trasformazione. È difficile che i dati estratti dalle sorgenti siano già pronti per essere sottoposti agli algoritmi di Data Mining, in quanto essi presentano inconsistenze, ridondanze e outliers[2]. Per gli attributi numerici si utilizzano misure statistiche come media e varianza, ma anche strumenti grafici come i box-plot e gli istogrammi; per gli attributi categorici si utilizzano come misure statistiche la moda e strumenti grafici come i grafici a torta. Nella fase di preprocessing è fondamentale gestire eventuali dati mancanti che possono essere causati da diversi motivi come la mancata registrazione da parte di chi effettua la raccolta dati oppure a degli errori nel sistema di raccolta dati stesso. Si può ricorrere a diverse soluzioni a seconda dell'analisi che si deve svolgere, per esempio è possibile:

- eliminare gli oggetti che contengono dati mancanti;
- ignorare i valori mancanti mentre si effettua l'analisi;
- effettuare una stima utilizzando la media e sostituire tale stima al posto del dato mancante;
- effettuare una previsione del dato mancante sulla base dei dati noti;
- sostituire il dato mancante con un valore deciso dall'analista.

Durante la fase di preprocessing, è di primaria importanza effettuare la pulizia dei dati o *data cleaning*. Lo scopo è quello di eliminare la ridondanza nei dati; tuttavia effettuare tale operazione sull'intero dataset è oneroso in termini di risorse computazionali e può richiedere molto tempo, per questo si sceglie un sottogruppo di oggetti rappresentativi del dataset originale e si esegue su di essi un campionamento.
Un altro problema da affrontare in questa fase è la riduzione della dimensionalità dei dati, in modo tale da ridurre la quantità di tempo e di memoria utilizzata dagli algoritmi di data mining. Esistono diversi modi per effettuare la riduzione della dimensionalità: citiamo a titolo di esempio la *Singular Value Decomposition (SVD)* e la *Principal Component Analysis (PCA)*. È possibile inoltre creare nuovi

[2]In statistica, il termine outlier definisce un valore anomalo e distante rispetto alle altre osservazioni disponibili

attributi che riassumano meglio le informazioni rilevanti: questi possono essere attributi creati a partire da quelli esistenti o trasformazioni su nuovi spazi.
Un'altra tecnica utile è quella di convertire attributi da dominio continuo a discreto, individuando il numero di intervalli più adatto: per fare ciò, è possibile ricorrere a tecniche supervisionate o non supervisionate.

2.3 Data Mining: estrazione della conoscenza

In questa fase del processo del KDD vengono applicati, sul dataset pulito i cui dati sono stati già trasformati, gli algoritmi di Data Mining con cui si vogliono identificare i pattern più significativi. Le tecniche di Data Mining si suddividono in tecniche descrittive, ovvero che effettuano l'estrazione dei modelli interpretabili a partire dai dati, oppure tecniche predittive ovvero che sfruttano alcune delle variabili note per predire valori incogniti o futuri di altre variabili.
Le tecniche di Data Mining vengono classificate in base al grado di intervento dell'analista e sono generalmente suddivise in due classi:

- **Tecniche supervisionate**: comprendono modelli in cui esistono una o più variabili indipendenti (input) che caratterizzano una o più variabili dipendenti (output). Tali strategie di apprendimento possono essere ulteriormente classificate sia in funzione della tipologia degli attributi di output (categorici o numerici), sia in funzione del fatto che i modelli siano stati progettati per effettuare un'interpretazione o una previsione;

- **Tecniche non supervisionate**: comprendono modelli che non ammettono una variabile dipendente, di conseguenza tutti gli attributi inclusi nel modello sono delle variabili indipendenti.

Nei prossimi paragrafi spieghiamo gli algoritmi utilizzati: algoritmi di clustering ed algoritmi di classificazione.

2.3.1 Gli algoritmi di clustering

Il *clustering* è considerato la principale tecnica di apprendimento non supervisionato. Questi algoritmi sono definiti come processi che hanno come obiettivo quello di raggruppare i record, contenuti in un dataset, in modo tale da definire degli insiemi che siano il più possibile omogenei. Eseguire un algoritmo di clustering significa quindi individuare delle classi di oggetti, chiamate *cluster*, in maniera che gli elementi contenuti in un cluster siano omogenei tra loro in base ai valori degli attributi che descrivono i record; inoltre tali algoritmi operano in modo che gli elementi che appartengono a cluster diversi siano disomogenei in base ai valori degli attributi che descrivono i record. In altre parole, un buon algoritmo di cluster massimizza

la similarità *intra-cluster* e minimizza la similarità *inter-cluster*, come possiamo notare in figura 2.2.

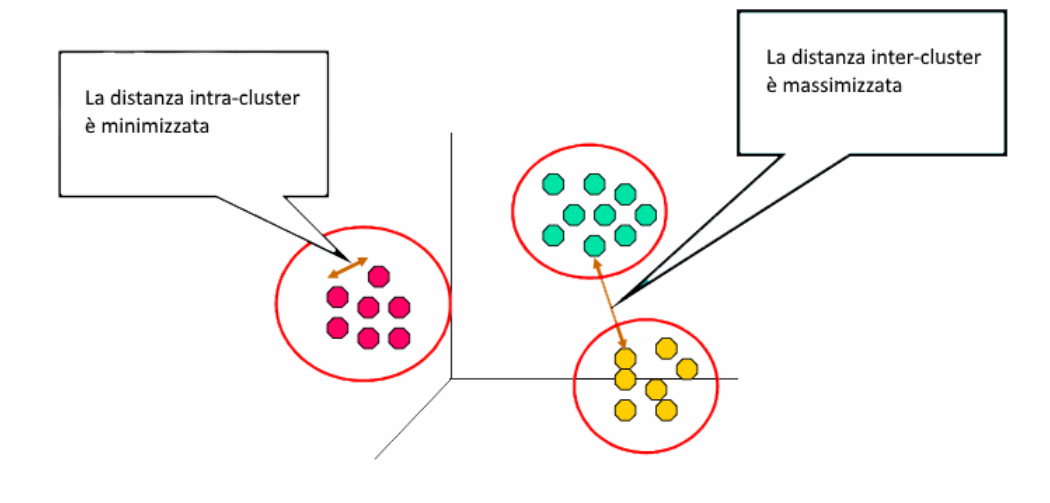


Figura 2.2: Creazione dei cluster in base alla similarità degli oggetti che contengono. © Tan, Steinbach, Kumar

Gli algoritmi di clustering sono esaustivi e mutuamente esclusivi: suddividono cioè in classi tutti i dati che fanno parte del dataset originario generando delle partizioni dell'insieme stesso. Esistono in realtà anche degli algoritmi non esclusivi, dove è possibile che un oggetto possa essere presente in cluster differenti con un grado di appartenenza diverso (*fuzzy clustering*). Le tecniche di clustering possono appartenere a due diverse filosofie:

- Bottom-up: l'inizializzazione dell'algoritmo prevede che gli oggetti vengano considerati tutti come cluster a sé stanti: avremo tanti cluster quanti sono gli oggetti presenti nel dataset. L'algoritmo provvede ad unire i cluster più vicini e continua ad operare in questo modo fin quando non viene raggiunto il numero prefissato di cluster o fin quando la distanza fra i cluster non superi una certa soglia;

- Top-down: l'inizializzazione dell'algoritmo prevede che gli oggetti vengano considerati in un cluster unico. La tecnica di clustering provvede a separare l'unico cluster in differenti cluster di dimensioni minori. Come nel caso precedente, l'algoritmo conclude il suo lavoro fin quando non viene raggiunto il numero prefissato di cluster o fin quando la distanza fra i cluster non superi una certa soglia.

Descritte le due filosofie a cui può appartenere una tecnica di clustering, dobbiamo eseguire un'ulteriore classificazione di tali algoritmi. Infatti, la letteratura distingue gli algoritmi di clustering in due categorie: algoritmi gerarchici e algoritmi partizionati.

Algoritmi gerarchici. Le tecniche di clustering appartenenti a questa categoria, generano una gerarchia di partizioni seguendo approcci di unione o di separazione, a seconda che si scelga una filosofia Bottom-up o Top-down. I risultati di una tecnica di clustering gerarchico vengono generalmente rappresentati da un dendogramma, una struttura ad albero simile a quella mostrata in figura 2.3. I tratti orizzontali del dendogramma sono chiamati nodi, mentre i tratti verticali vengono chiamati internodi. La distanza che intercorre fra un nodo del dendogramma e la sua base è proporzionale alla similarità fra due o più oggetti di cui il nodo stesso rappresenta la fusione. Un vantaggio significativo del clustering gerarchico è che una tecnica di questo tipo non ha bisogno di conoscere preventivamente il numero desiderato di cluster: questo è ottenibile tagliando il dendogramma al livello appropriato. Lo svantaggio principale è la rigidità: una volta effettuata la separazione o l'unione di due o più oggetti, non è più possibile tornare alla situazione precedente e correggere un eventuale errore.

Figura 2.3: Clustering gerarchico tradizionale con dendogramma.© Tan, Steinbach, Kumar

Algoritmi partizionati. Gli algoritmi partizionati, altresì detti non gerarchici, agiscono con una logica diversa. Considerando un dataset avente al suo interno n record, tali algoritmi costruiscono K diverse partizioni, con $1 < K \leq n$, in cui ogni partizione stessa rappresenta un cluster. Tali tecniche classificano i dati in K diversi insiemi, in modo che ogni insieme deve contenere almeno un dato ed ogni dato deve appartenere ad un solo insieme. É possibile che, in alcuni casi particolari, un record può appartenere a più di un insieme. Il numero K viene definito come input dell'algoritmo. Questo crea una prima partizione; successivamente l'algoritmo esegue una procedura iterativa che muove i record da un cluster all'altro a seconda del criterio utilizzato per valutare la similarità degli oggetti. Per ottenere la soluzione ottima globale attraverso le tecniche di clustering con partizioni, occorre enumerare tutte le partizioni possibili. Proprio a causa di ciò, la maggior parte degli algoritmi

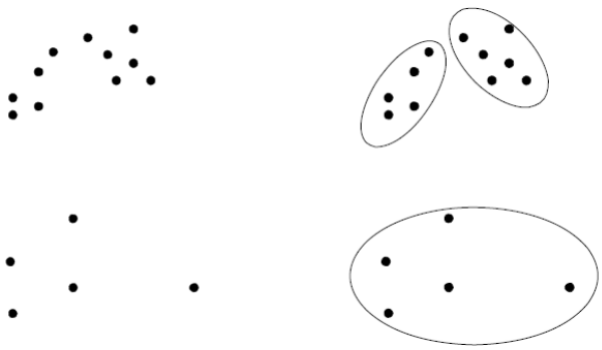


Figura 2.4: Clustering partizionato.© Tan, Steinbach, Kumar

di clustering partizionato utilizzano metodi euristici in cui ogni cluster viene considerato come un punto che, a seconda delle scelte progettuali, può coincidere con il valore medio degli oggetti del cluster, può essere uno dei punti che si trovano in prossimità del centro del cluster, o in alternativa un punto definito con metodi simili. Gli algoritmi di clustering che utilizzano metodi euristici funzionano quando i cluster sono di forma sferica ed i database hanno dimensioni contenute; in caso contrario è necessario utilizzare estensioni di queste tecniche.

La distanza tra cluster. Una decisione fondamentale da prendere inizialmente è quella della misura della distanza, che consente di traslare nel campo numerico il concetto di similarità e dissimilarità tra elementi che fanno parte di cluster differenti.

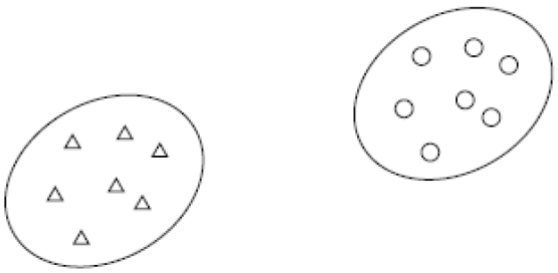


Figura 2.5: Clustering: la distanza intra-cluster è decisamente minore rispetto a quella inter-cluster.© Dulli, Furini, Peron

Per esprimere il concetto di distanza, possiamo immaginare ogni record appartenente al dataset come se fosse un punto in uno spazio multidimensionale. Supponiamo,

come in figura 2.5, che lo spazio in cui si dispongono i vari record sia bidimensionale e che gli oggetti che presentano caratteristiche simili vengano rappresentati da simboli uguali. É facile riconoscere nella figura 2.5 due cluster, in cui la distanza fra gli oggetti appartenenti allo stesso cluster è minore rispetto alla distanza tra oggetti che appartengono a cluster differenti.
Nella presente tesi, la distanza utilizzata negli algoritmi di clustering è sempre la distanza euclidea. Ricordiamo brevemente che cosa intendiamo per distanza e similarità.

Indichiamo con S la rappresentazione simbolica di uno spazio e siano x, y, z tre punti qualsiasi di S. La *distanza* è una funzione $d(x, y) : S \times S \longrightarrow \mathbb{R}$ che soddisfa le seguenti condizioni:

1. $d(x, y) \geq 0 \ \forall x, y \in S$;
2. $d(x, y) = 0 \Leftrightarrow x = y$;
3. $d(x, y) = d(y, x) \ \forall x, y \in S$;
4. $d(x, y) \leq d(x, z) + d(z, y) \ \forall x, y, z \in S$.

Le prime tre condizioni richiedono rispettivamente che la distanza sia una funzione non nulla, riflessiva e simmetrica. L'ultima condizione è meno intuitiva rispetto alle precedenti: questa disuguaglianza è nota come *disuguaglianza triangolare*, richiede che la distanza fra due punti x e y sia minore o uguale alla somma delle distanze tra i due punti ed un punto z diverso dai precedenti. Sia S la rappresentazione

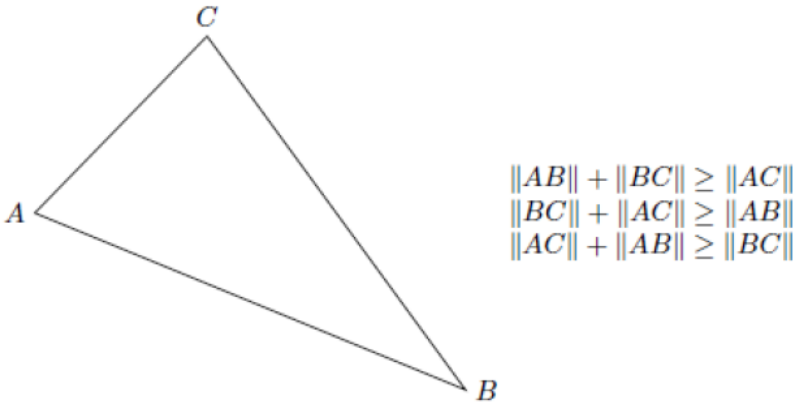


Figura 2.6: Disuguaglianza triangolare: significato geometrico. © Dulli, Furini, Peron

simbolica di uno spazio geometrico ed x, y due punti appartenenti ad esso. Una funzione $w(x, y) : S \times S \longrightarrow \mathbb{R}$ che soddisfa le seguenti condizioni:

1. $s(x, y) = 1 \Leftrightarrow x = y$;
2. $s(x, y) = s(y, x)\ \forall x, y \in S$.

Le due condizioni indicano rispettivamente quando vi è presenza di massima similarità e che inoltre tale funzione gode della proprietà di simmetria.
Ora che i concetti di distanza e di similarità sono stati richiamati, possiamo riportare la definizione matematica di distanza euclidea. Siano X e Y due vettori di lunghezza n:

$$X = (x_1, x_2, \ldots, x_i, \ldots, x_n);$$

$$Y = (y_1, y_2, \ldots, y_i, \ldots, y_n);$$

definiamo distanza euclidea tra X e Y la seguente funzione:

$$D(X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

2.3.2 Metodi di classificazione: gli alberi di decisione

Per metodi di classificazione si intendono gli algoritmi che prevedono l'assegnazione di un nuovo oggetto ad una classe predefinita dopo averne esaminato le caratteristiche. Gli alberi di decisione costituiscono il metodo più semplice per la classificazione dei pattern in un numero finito di classi. Tali algoritmi portano alla costruzione di un albero, dove i sottoinsiemi di record vengono chiamati nodi; i sottoinsiemi finali in cui vengono raggruppati gli oggetti vengono chiamate foglie.
I nodi sono etichettati con i nomi degli attributi, i rami dell'albero invece sono etichettati con i valori che lo specifico attributo può assumere. La classificazione dell'oggetto si può individuare percorrendo l'albero dalla radice ad una delle foglie. I percorsi rappresentati dall'albero forniscono una serie di regole.
Per capire come opera un albero di decisione, dobbiamo innanzitutto introdurre i concetti di*entropia* ed *information gain*. Consideriamo un problema di classificazione in cui sono presenti due sole classi che chiamiamo + e −, S è l'insieme dei record con cui dobbiamo creare un albero. Indichiamo con P_+ la percentuale di record classificati con + e con P_- la percentuale di record classificati con −. L'entropia di S è una funzione definita dalla seguente espressione:

$$H(S) = -P_+ log_2 P_+ - P_- log_2 P_-$$

Dal grafico presente in figura 2.7 notiamo come l'entropia sia una funzione compresa tra 0 ed 1, in particolare $H(S) = 0$ se $P_+ = 100\%$ o $P_- = 100\%$, ovvero nel caso in cui tutti gli oggetti sono classificati come appartenenti ad una delle due classi; $H(S) = 1$ si verifica nel caso in cui $P_+ = 50\%$ e $P_- = 50\%$, ovvero nel caso in cui gli oggetti siano divisi equamente fra le due classi.

L'entropia misura l'ordine dello spazio dei record considerato per la costruzione dell'albero. Un valore di entropia prossimo all'unità, indica un elevato disordine dello spazio dei record, ciò significa che è elevata la difficoltà che l'algoritmo incontra nell'assegnare ogni record alla propria classe sulla base delle caratteristiche che lo definiscono: più è elevata l'entropia e meno informazioni abbiamo a disposizione sull'attributo classe.

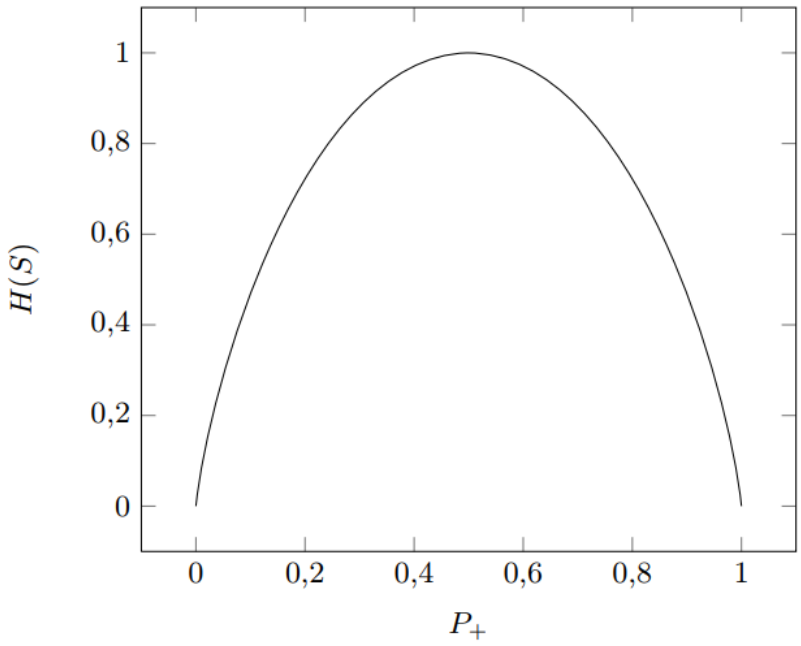


Figura 2.7: Entropia di S.© Dulli, Furini, Peron

In generale, partendo da una situazione di massimo disordine, ovvero $H(S) = 1$, una partizione dei record effettuata in base all'attributo X porterebbe il sistema ad un nuovo valore di entropia minore rispetto a quello di partenza. In quest'ottica definiamo l'*information gain* come la diminuzione di entropia che otteniamo partizionando i dati rispetto ad un attributo X. Se $H(S)$ è il valore iniziale di entropia ed $H(S, X)$ è il valore dell'entropia dopo che l'algoritmo ha effettuato la partizione dei record con l'attributo X, allora l'*information gain*, indicato con G, è dato da:

$$G = H(S) - H(S, X).$$

Questa grandezza è tanto più elevata quanto maggiore è la diminuzione di entropia dopo che gli oggetti sono stati partizionati con l'attributo X. Pertanto, è chiaro che un criterio di scelta dei nodi di un classificatore ad albero consiste nello scegliere, ad ogni iterazione, l'attributo X che massimizza l'*information gain*. Questa caratteristica presenta valori elevati in corrispondenza degli attributi che sono altamente informativi e che aiutano a classificare con buona probabilità la classe di

appartenenza degli oggetti. Occorre però fare una precisazione: spesso gli attributi ad alto contenuto informativo difficilmente sono generalizzabili. Per esempio, considerando un dataset estratto da una sorgente di dati di proprietà di una banca, il codice fiscale di un cliente è molto informativo, poiché identifica con certezza il cliente stesso, ma non può essere generalizzabile. Occorre individuare gli attributi che sono ad alto contributo informativo ed allo stesso tempo generalizzabili.

Algoritmo ID3. Un classificatore ad albero viene costruito tramite un procedimento ricorsivo, in cui in ogni passo dell'algoritmo si utilizzano tecniche euristiche o statistiche, con l'obiettivo di individuare quale attributo inserire nel nodo. Questo tipo di algoritmo lavora solo su attributi nominali, ne consegue che gli attributi continui devono essere prima discretizzati. Viene utilizzato un approccio top-down e si applica ricorsivamente la tecnica *divide et impera*. Le fasi dell'algoritmo sono rappresentate di seguito.

Algoritmo ID3

input: samples
attribute list
output: decision tree

1: crea un nodo N
2: **if** *samples sono tutti nella stessa classe C* **then**
3: return N come foglia etichettata con la classe C
4: **end if**
5: **if** *attributi sono vuoti* **then**
6: return N come foglia etichettata con la classe più comune
7: **end if**
8: seleziona l'attributo con miglior information gain
9: etichetta il nodo N con *test - attribute*
10: **for** *valore conosciuto a_i dell'attributo* **do**
11: costruisci un arco dal nodo N per la condizione attributo = a_i s_i= insieme delle tuple nel training che soddisfano la condizione a_i
12: **if** s_i è vuoto **then**
13: attacca una foglia etichettata con la classe più comune
14: **else**attacca il nodo ritornato da *Genera un albero di decisione (s_i, list)*
15: **end if**
16: **end for**

L'algoritmo è inizializzato con un nodo che rappresenta il training set (passo 1). Se le istanze fanno parte della stessa classe, il nodo diventa un'unica foglia avente per etichetta la classe alla quale appartengono tutti gli oggetti (passi 2-5). Altrimenti,

l'algoritmo utilizza l' *information gain* come metodo euristico per selezionare l'attributo che separerà meglio gli oggetti in classi individuali. Tale attributo prenderà il nome di *decision attribute*.
Per ogni valore conosciuto dal *test attribute*, viene creato un arco che partizionerà l'insieme degli oggetti. L'algoritmo itera questo metodo sugli oggetti, escludendo dall'insieme degli attributi quelli utilizzati nello split corrente. L'algoritmo termina se tutti gli oggetti di un nodo appartengono alla stessa classe, oppure se non ci sono attributi ulteriori con cui effettuare nuove partizioni o ancora se non ci sono oggetti per l'arco che è stato individuato come *test attribute*.
Costruito il classificatore ad albero, molti analisti ne eseguono una potatura. L'obiettivo della potatura è quello di eliminare alcuni rami e nodi che causano overfitting nei dati. Con la potatura, vengono sostituiti alcuni nodi interni con delle foglie e vengono rimossi alcuni rami dell'albero che non sono raggiungibili dalla radice. Nella presente tesi, non è stato utilizzato alcun metodo di potatura. Tuttavia, segnaliamo che in letteratura esistono differenti metodi, sia euristici che analitici.

2.4 Interpretazione e valutazione della conoscenza

Una volta che i pattern sono stati individuati ed estratti grazie alle tecniche di Data Mining, l'analista, eventualmente con l'ausilio di un esperto di dominio, analizza i risultati valutando la correttezza e la qualità delle informazioni ricavate. Questa fase è molto onerosa in termini di costi temporali, in quanto spesso si iterano i punti precedenti del processo di *KDD* per migliorare il modello.

Capitolo 3

Ambiente di sviluppo

In questo capitolo descriveremo gli ambienti di sviluppo che sono stati utilizzati per svolgere la presente tesi. In particolare, la scelta è ricaduta su due software *open source* che si prestano bene all'utilizzo delle tecniche di Data Mining: RStudio e RapidMiner.

3.1 RStudio

RStudio è un *integrated development environment (IDE)*, gratis ed open source, utile per programmare con il linguaggio di programmazione R. Questo, è utilizzato particolarmente per calcoli statistici e per l'implementazione di tecniche di Data Mining. É molto valido anche per la gestione dei dati e per la loro produzione. Ha la caratteristica di essere disponibile in due edizioni: RStudio Desktop, dove il programma è eseguibile localizzato fra le applicazioni presenti nel desktop del nostro pc e RStudio Server, che permette l'accesso ad RStudio utilizzando un web browser mentre il software è in esecuzione su un server remoto Linux. RStudio è disponibile per Windows, Linux e macOS per la versione desktop. Nella presente tesi è stata utilizzata la versione desktop per Windows 10. Nella figura 3.1 possiamo vedere come si presenta l'ambiente RStudio.
In particolare, possiamo suddividere l'ambiente di sviluppo in quattro diverse aree:

1. *Code Editor*: in questa sezione avviene la stesura del codice R con l'apertura, la creazione e l'implementazione degli script;

2. *R Console*: in questa sezione vengono eseguiti i comandi R. Durante la fase di esecuzione, R mostrerà all'utente eventuali errori proprio in quest'area;

3. *Workspace and History*: in questa sezione vengono elencati gli oggetti che sono stati creati, per esempio variabili, data frame, tabelle, ecc. É possibile rimuovere gli oggetti che non sono più utilizzati, in quanto RStudio salva e mantiene

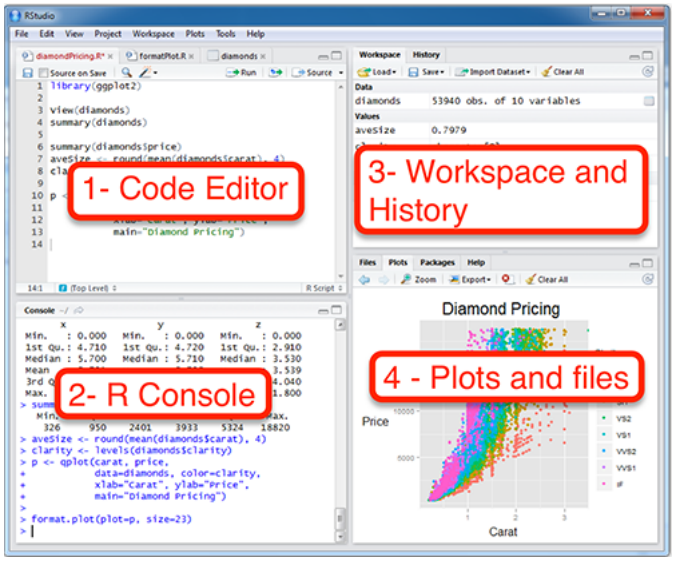


Figura 3.1: Ambiente di sviluppo RStudio

automaticamente tutti gli oggetti creati anche quando questi vengono rimossi dalla sezione *Code editor*;

4. *Plots and files*: in questa sezione è possibile caricare pacchetti, aprire i file di *help* di R e visualizzare i plot che sono stati creati.

RStudio è dotato di molte librerie installate di default. Tuttavia, tramite il sito *CRAN (Comprehensive R Archive Network)* è possibile scaricare altri packages, in base alle esigenze di chi progetta. All'interno dei package, esiste una grande quantità di funzioni utilizzabili sfruttando il linguaggio di programmazione R. Essendo un IDE open source, ogni mese è possibile accedere a nuovi packages scritti e rilasciati da ricercatori ed ingegneri. Elenchiamo di seguito i packages più importanti che sono stati scaricati dal sito *CRAN*.

- **broom**: converte gli oggetti utilizzati per l'analisi statistica su R in tabelle ordinate di dati, in modo che i dati stessi possano essere facilmente combinati, rimodellati ed elaborati con strumenti. Il pacchetto fornisce tre generiche funzioni: *tidy*, che riassume i risultati statistici di un modello, per esempio i coefficienti di una regressione; *augment*, che aggiunge colonne ai dati originali, le nuove colonne possono essere previsioni, residui e assegnazioni di cluster; *glance*, provvede ad inserire una riga di statistiche riepilogative del modello;

- **knitr**: progettato per essere un motore di generazione di report dinamici. Combina in un pacchetto molte funzionalità che erano collocate in packages differenti;

- **dbscan**: pacchetto che contiene funzioni utili per una rapida implementatone degli algoritmi di clustering basati sulla densità. Include il DBSCAN (clustering basato sulla densità) e OPTICS (ordinamento per identificare la struttura dei cluster) oltre che HDBSCAN (DBSCAN gerarchico) e l'algoritmo LOF (Local Outliers Factor). Le implementazioni utilizzano la struttura dei dati kd-tree per eseguire velocemente l'algoritmo di ricerca del *K-Nearest Neighbour (KNN*);

- **grid**: pacchetto che contiene un sistema grafico che integra la grafica di default di R;

- **sqldf**: la funzione più importante di questo pacchetto è *sqldf()*, passata ad un singolo argomento rappresentante una query SQL, in cui i nomi delle tabelle sono i dataframes creati in R. La funzione *sqldf()* imposta in modo trasparente un database, importa i dataframe in quel database, esegue la query SQL e restituisce il risultato nella *R Console*;

- **NBClust**: questo package consente di utilizzare 30 indici diversi per determinare il numero ottimale di cluster e propone all'utente il miglior schema di clusterizzazione in base ai risultati ottenuti, esplorando tutte le combinazioni dei numeri di cluster, misure di distanza e metodi di clustering;

- **mctest**: tale pacchetto contiene delle funzioni che rappresentano delle misure diagnostiche della multicollinearità, un concetto teorico collegato alla Regressione lineare multipla. Nel pacchetto, sono presenti funzioni che indicano quali regressori possono essere la ragione della collinearità tra regressori stessi.

3.2 RapidMiner

RapidMiner è una piattaforma software di *data science* sviluppata dall'omonima compagnia che fornisce un ambiente integrato per la preparazione dei dati, il *machine learning*, il *deep learning*, il *text mining* e l'analisi predittiva. Viene utilizzato per applicazioni commerciali, per ricerca, istruzione, formazione, prototipazione rapida e sviluppo di applicazioni; supporta tutte le fasi del processo di apprendimento automatico, compresa la preparazione dei dati, la visualizzazione dei risultati, la convalida del modello e l'ottimizzazione. RapidMiner è sviluppato su un modello *open core*. La versione gratuita di RapidMiner Studio, che è limitata ad un processore logico e 10.000 righe di dati, è disponibile con l' *Affero General Public License (AGPL)*.
RapidMiner utilizza un modello *client/server*, con il server offerto come *on-premise* o in infrastrutture cloud pubbliche e private. RapidMiner fornisce il 99% di una

soluzione analitica avanzata[1] attraverso framework basati su template che velocizzano l'esecuzione e riducono gli errori, eliminando quasi la necessità di scrivere in codice. RapidMiner offre procedure di data mining e machine learning che includono: processi di ETL, preprocessing e visualizzazione dei dati, analisi predittiva, modellazione statistica, valutazione e implementazione. RapidMiner è scritto tramite linguaggio di programmazione Java. RapidMiner fornisce una *Graphical User Interface (GUI)* per progettare ed eseguire flussi di lavoro analitici. Questi flussi di lavoro sono chiamati *processi* in RapidMiner e sono costituiti da più *operatori*. Ogni operatore esegue una singola attività all'interno del processo e l'output di ciascun operatore costituisce l'input di quello successivo. In alternativa, il motore può essere richiamato da altri programmi o utilizzato come API. Le singole funzioni possono essere richiamate dalla riga di comando. RapidMiner fornisce schemi di apprendimento, modelli e algoritmi e può essere esteso utilizzando script R e Python.
La funzionalità RapidMiner può essere estesa con plug-in aggiuntivi (*Weka Extension*, *Text Processing*,ecc.) resi disponibili tramite *RapidMiner Marketplace*. Il Marketplace di RapidMiner offre agli sviluppatori una piattaforma per creare algoritmi di analisi dei dati e pubblicarli nella comunità.

[1]Fonte: Bloor Research

Capitolo 4

Il framework F-SCAN

Il sistema che è stato progettato per l'estrazione dei pattern dalle certificazioni energetiche è riassunto in figura 4.1. Il framework prende il nome di F-SCAN (**F**eatures-**S**election for **C**ertificates **AN**alysis) ed è composto da quattro blocchi principali, che sono: *Raccolta e integrazione dei dati, Processo di ottimizzazione, Processo di analisi ed estrazione della conoscenza, Esplorazione della conoscenza.*

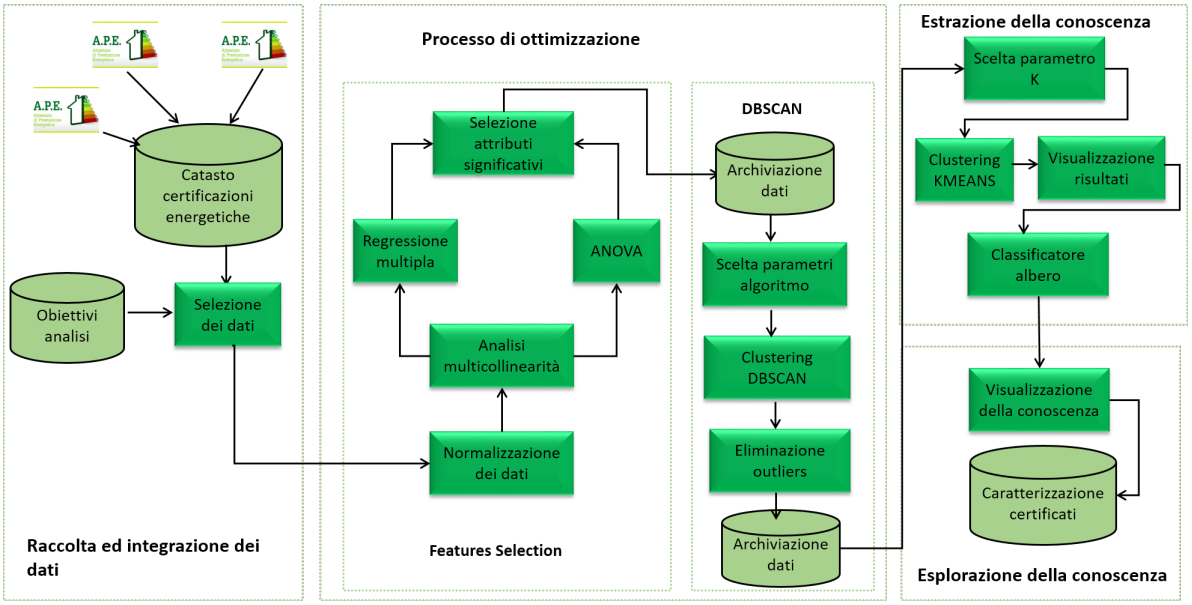


Figura 4.1: L'architettura F-SCAN

La *Raccolta e integrazione dei dati* estrae dal dataset iniziale i dati relativi alle certificazioni energetiche, provenienti dall'apposito Catasto della Regione Piemonte. Tali dati vengono integrati con i valori limite che sono fissati dalle normative in vigore nel momento in cui la particolare certificazione è stata emessa; ulteriori integrazioni vengono operate con le indicazioni che sono reperibili nel manuale

utilizzato dal tecnico che compila la certificazione energetica. La fase successiva è quella del *Processo di ottimizzazione*, in cui viene effettuata la *Feature Selection*, che consente di eliminare gli attributi statisticamente irrilevanti, ed il *DBSCAN*, un algoritmo di clustering basato sulla densità, sfruttato per rimuovere gli *outliers* dal dataset. Nel *Processo di analisi ed estrazione della conoscenza*, viene applicato l'algoritmo di clustering *K-Means*, seguito dall'utilizzo del *Classificatore ad albero*. Infine, l' *Esplorazione della conoscenza* viene eseguita sfruttando l'utilizzo di differenti tipologie di grafico, con l'obiettivo di mostrare in maniera chiara e semplice la caratterizzazione delle Certificazioni energetiche. Prima di proseguire con la spiegazione dei blocchi del framework occorre fare una precisazione. Poiché la presente tesi si avvale di un lavoro di tesi effettuato precedentemente, non verrà trattata la fase di *preprocessing* dei dati. Non è stato necessario pensare a come trattare i dati mancanti, perché non ne erano presenti nel nostro dataset iniziale. Nelle prossime sezioni, verranno descritti nel dettaglio i blocchi dell'architettura F-SCAN.

4.1 Raccolta ed integrazione dei dati

Il dataset utilizzato come punto di partenza della presente tesi include una parte delle certificazioni energetiche contenute nel Catasto energetico degli edifici della Regione Piemonte, raccolte in un intervallo di tempo compreso tra il 1 luglio 2009 e il 30 giugno 2014, più precisamente quelle compilate in base alle direttive dettate dalla normativa tecnica UNI/TS 1300:2008. In totale, il dataset si compone di 533.959, raggruppati per semestri in file Excel, e di 101 attributi. Chiaramente, abbiamo utilizzato solo una piccola parte degli attributi contenuti nel dataset iniziale. La selezione delle caratteristiche rilevanti ai fini della nostra analisi è stata effettuata con la collaborazione degli esperti di dominio appartenenti al Dipartimento Energia del Politecnico di Torino, che hanno suggerito quali attributi, fra i tanti contenuti nel dataset di partenza, influiscono sulle prestazioni energetiche degli edifici.
All'interno di questa fase sono stati calcolati nuovi attributi di sintesi, contenenti indici e rendimenti corretti con i Gradi Giorno, in modo da misurare la localizzazione teorica degli edifici a Torino; alcune caratteristiche invece sono state aggregate ad altre per riuscire ad estrapolare informazioni aggiuntive. Questi nuovi attributi sono stati inclusi nel dataset. Considerando gli obiettivi della nostra analisi, riportiamo gli attributi che sono stati selezionati. Per la loro descrizione, si rimanda all'apposita sezione 1.2.

- Volume lordo riscaldato
- Superficie disperdente totale
- Superficie utile
- Altezza media

- Fattore forma
- Trasmittanze opache
- Trasmittanze trasparenti
- Rendimento di generazione decimale
- Rendimento di distribuzione decimale
- Rendimento di regolazione decimale
- Rendimento di emissione decimale
- Rendimento globale riscaldamento Torino
- Rendimento globale acqua calda sanitaria
- Rendimento globale riscaldamento acqua calda sanitaria
- Rendimento stagionale acqua calda sanitaria
- Rendimento medio globale stagionale acqua calda sanitaria
- Fabbisogno energia termica utile
- Fabbisogno energia termica utile acqua calda sanitaria
- Fabbisogno energia termica utile Torino
- Fabbisogno acqua calda sanitaria soddisfatto da fonti rinnovabili
- Indice di prestazione energetica acqua calda sanitaria Torino
- Indice di prestazione del riscaldamento Torino
- Indice di prestazione globale energetica Torino
- Indice di prestazione energetica acqua calda sanitaria fonti rinnovabili Torino
- Prestazione energetica acqua calda sanitaria check
- Potenza riscaldamento
- Prestazione raggiungibile
- Classe energetica

4.2 Processo di ottimizzazione

Il processo di ottimizzazione è composto da due sottoblocchi: la Feature Selection ed il DBSCAN. Per poter individuare pattern significativi dal dataset mantenendo la numerosità delle relazioni entro limiti gestibili, l'analisi deve essere effettuata solamente su un sottoinsieme ristretto del dataset iniziale. La selezione degli attributi, in via preliminare a qualsiasi tipo di analisi, è di vitale importanza per riuscire ad estrarre la conoscenza nascosta. Per effettuare la selezione degli attributi, l'F-SCAN utilizza la regressione lineare multipla e l'*ANOVA*; per l'individuazione degli outliers, invece il framework sfrutta il DBSCAN, in modo da separare il rumore dai dati rilevanti per il nostro studio. Prima ancora di poter applicare le due tecniche, il dataset è stato normalizzato. Dopo che sono stati individuati gli attributi rilevanti, abbiamo applicato il K-Means per effettuare il clustering e successivamente il classificatore ad albero per la *cross-validation*, utilizzando come *label* l'etichetta di cluster prima e la classe energetica poi. Le informazioni ottenute sono state poi visualizzate attraverso grafici di sintesi.

Normalizzazione. Prima di eseguire la Features Selection è stata effettuata la normalizzazione dei dati. Questa è una trasformazione che viene operata sui dati stessi, in modo che essi siano confrontabili e rientrino in intervalli stabiliti. L'obiettivo della normalizzazione è quello di trasformare i valori assunti dagli attributi dei dati, affinché essi assumano determinate proprietà. Se da un'analisi preliminare notiamo che le variabili mostrano andamenti diversi e se c'è una grande differenza di valori fra le variabili stesse, allora occorre eseguire la trasformazione. Riportiamo di seguito le normalizzazioni che sono state utilizzate nella presente tesi:

- Normalizzazione Z-Score: dato un insieme di attributi, chiamiamo μ_x la loro media e σ_x la deviazione standard. La normalizzazione Z applica ai dati la seguente trasformazione:

$$x' = \frac{x - \mu_x}{\sigma_x}$$

  Tale trasformazione crea una variabile con media nulla e deviazione standard pari a 1. Una normalizzazione Z-Score viene utilizzata specialmente quando non si conoscono il minimo ed il massimo per una determinata caratteristica.

- Normalizzazione min-max: questo tipo ti trasformazione scala i valori di un attributo X in modo che essi cadano in un nuovo intervallo $[new_{min}, new_{max}]$. Per applicare la normalizzazione min-max, si utilizza la seguente formula:

$$x' = \frac{(x - min_x)}{(max_x - min_x)}(new_{max} - new_{min}) + new_{min}$$

La min-max è un tipo di normalizzazione molto influenzata dagli outliers. Nel caso delle certificazioni energetiche, i valori attribuiti al nuovo minimo ed al nuovo massimo sono stati rispettivamente 0 e 1. In questo modo, abbiamo mantenuto la positività dei dati di partenza, che essendo delle caratteristiche termiche e fisiche non possono mai essere minori di zero.

4.2.1 Features selection

In generale, un problema fondamentale del *Machine Learning* è la *Features Selection*, nota anche come *Variable Selection* o *Attribute Selection*. L'obiettivo principale della Features Selection è la riduzione della dimensionalità di un determinato dataset. Se avessimo dei dispositivi con potenza di calcolo e memoria sufficienti, vorremmo poter utilizzare tutte le dimensioni del dataset, incluse le feature irrilevanti, così da approssimare meglio le relazioni sottostanti tra gli input e gli output del nostro modello. Tuttavia, ci sono due problemi che possono insorgere a causa della presenza delle caratteristiche irrilevanti coinvolte nel processo di apprendimento:

1. Le caratteristiche irrilevanti generano grandi costi computazionali. Diversi esperimenti hanno infatti evidenziato che il costo computazionale cresce linearmente con l'aumentare del numero delle feature; inoltre se il numero delle previsioni da eseguire è elevato, la ricerca dimostra che il costo computazionale cresce esponenzialmente con l'aumentare del numero delle caratteristiche.

2. Le caratteristiche irrilevanti possono generare *overfitting*. Per esempio, nel campo delle diagnosi mediche, lo scopo è quello di stabilire una relazione tra i sintomi che si manifestano nel paziente e la loro corrispondente diagnosi. Se venisse incluso il numero identificativo del paziente come input del processo di Machine Learning, questo potrebbe giungere alla conclusione che la malattia è determinata proprio dall'ID del paziente stesso.

Inoltre, dal momento che il nostro obiettivo è quello di approssimare la relazione sottostante tra input ed output, è ragionevole ignorare gli input che abbiano scarse relazioni con l'output, in modo da mantenere ridotte le dimensioni del modello di approssimazione. Un metodo che si potrebbe utilizzare per la Features Selection è il metodo brute-force, che consiste nel valutare esaustivamente tutte le combinazioni possibili di attributi da utilizzare come input del modello, in modo poi da trovare il sottoinsieme ottimale. Il costo computazionale di questo metodo è ovviamente proibitivo. Pertanto, si ricorre ad algoritmi greedy, ovvero dei metodi che cercano di ottenere una soluzione ottima attraverso la scelta della migliore soluzione ad ogni passo locale. Questa tecnica, spiegata passo dopo passo nella pagina seguente, consente di trovare soluzioni ottimali per risolvere determinati problemi in un tempo ragionevole. Per questo motivo, è stato scelto di utilizzare un algoritmo greedy che

Algoritmo Regressione Lineare per Features Selection

1: Selezione del dataset su cui effettuare la Features Selection
2: Selezione preliminare degli attributi:
3: **if** *Tipo attributo='numeric'* **then**
4: Includere attributo nella selezione
5: **else if** *Peso attribuibile ad attributo di tipo char* **then**
6: Attribuire peso ed includerlo nella selezione
7: **else**
8: Scartare attributo
9: **end if**
10: Normalizzazione del dataset
11: Calcolo per ogni dimensione della varianza:
12: **if** $\sigma^2 = 0$ **then** Scartare la dimensione
13: **else**
14: Mantenere la dimensione nell'analisi
15: **end if**
16: Analisi multicollinearità preliminare con decomposizione di Cholesky:
17: **if** *collinearità* $=' TRUE'$ **then**
18: Eliminare attributo
19: **else**Passare alla fase successiva
20: **end if**
21: Analisi overall multicollinearità
22: **while** *collinearità*$Regressore_i =' TRUE'$ **do**
23: Analisi individual regressor multicollinearità
24: Eliminazione delle dimensioni collineari, a partire da quelle con le misure diagnostiche più elevate
25: **end while**
26: Costruzione modello regressione:
27: Scelta della variabile dipendente e dei regressori;
28: Calcolo Regressione Lineare Multipla (RLM) e Analysis Of Variance (ANOVA);
29: Analisi misure di performance della RLM e dell'ANOVA a livello di singole variabili;
30: Eliminazione delle variabili che con RLM e con ANOVA hanno scarse misure di performance;
31: Se uno dei due metodi continua a rilevare variabili non rilevanti, procedere con la loro eliminazione per osservare se il modello migliora;
32: Ripetizione del procedimento dal punto 6b fin quando le misure di performance non rilevano che tutte le caratteristiche siano rilevanti.
33: Mantenere nel dataset il sottoinsieme di dimensioni che risultano essere rilevanti.
34: Mantenere nel dataset il sottoinsieme delle caratteristiche rilevanti

sfrutta la Regressione Multipla Lineare. Richiamiamo i concetti teorici utilizzati per implementare la Features Selection.

La Regressione Lineare.
La teoria della regressione formalizza il problema di una relazione funzionale tra le variabili misurate in base ai dati campionari che vengono estratti da una popolazione, la quale per ipotesi è di numerosità infinita. Oggi, in statistica, l'analisi della regressione è associata alla risoluzione di un modello lineare. Le tecniche di regressione sono molto versatili e possono essere impiegate nel campo delle scienze applicate ma anche in quello delle scienze sociali. Tali tecniche possono essere utilizzate solamente su dati numerici. La regressione lineare esegue una stima del valore atteso di una variabile dipendente che chiamiamo Y, dati i valori di altre variabili indipendenti chiamate regressori. Nel nostro caso di studio, per la Features Selection, abbiamo utilizzato la regressione lineare multipla. Il modello di regressione lineare multipla è:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_k X_{ki} + \mu_i$$

dove:

- i varia tra le osservazioni $i = 1, \ldots, n$;
- Y_i è la variabile dipendente;
- $X_{1i}, X_{2i}, \ldots, X_{ki}$ sono le i-esime osservazioni associate ai k regressori;
- $\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_k X_{ki}$ è la retta di regressione;
- β_k è il coefficiente angolare di X_k quando gli altri regressori vengono mantenuti costanti;
- μ_i è l'errore statistico.

Tale modello, possiede le assunzioni dei minimi quadrati (*Ordinary Least Squares*). Le assunzioni dei minimi quadrati per la regressione multipla sono:

- L'errore statistico μ_i ha una media condizionata nulla dati i regressori $X_1, X_2, \ldots, X_k$, ossia $E(\mu_i \mid X_{1i}, X_{2i}, \ldots, X_{ki}) = 0$;
- I regressori sono indipendenti e identicamente distribuiti (i.i.d.) dalla loro distribuzione congiunta;
- I regressori hanno momenti quarti finiti non nulli. Questa assunzione limita la probabilità di selezionare un'osservazione con valori estremamente elevati di X_i e di μ_i. In termini matematici, questo vuol dire che $0 < E(X_i^4) < \infty$ e $0 < E(\mu_i^4) < \infty$;

- Non vi è collinearità perfetta tra i regressori.

Raggruppando le osservazioni delle variabili di regressione in una matrice $\mathbf{X}$ di dimensioni $N \times (k+1)$, che si ipotizza avere rango pieno e uguale a $k+1$ (il termine costante, o intercetta, corrisponde ad avere una colonna di 1 nella $\mathbf{X}$), è possibile scrivere in notazione matriciale:

$$y = \mathbf{X}\beta + \epsilon$$

Nella formulazione più elementare, si assume che $\epsilon \sim N(0, \sigma^2 I)$, ossia: $E[\epsilon_i] = 0$ e $E[\epsilon_i^2] = \sigma^2 \forall$ (omoschedasticità), $E[\epsilon_i, \epsilon_j] = 0 \forall j \neq i$ (assenza di correlazione nei disturbi). Si ipotizza inoltre che:

$$E[\mathbf{X}'\epsilon_i] = 0$$

ciò vuol dire che non esiste correlazione fra i regressori ed i disturbi casuali. Questa ipotesi riveste un'importanza cruciale, poiché rende possibile considerare i regressori compresi nella matrice $\mathbf{X}$ come variabili esogene. Quest'ultima proprietà è tutt'altro che banale, in quanto soltanto laddove essa è valida è possibile garantire che il vettore delle stime dei parametri del modello $\hat{\beta}$ abbia per valore atteso il valore dei parametri β, godendo in questo modo della proprietà di correttezza. Sotto tali ipotesi, è possibile ottenere le stime del vettore di parametri β tramite il modello dei minimi quadrati, risolvendo il problema di minimo:

$$min_{\hat{\beta}}(y - \mathbf{X}\hat{\beta})'(y - \mathbf{X}\hat{\beta})$$

Le condizioni del primo ordine per un minimo definiscono il sistema:

$$-2\mathbf{X}'y + 2\mathbf{X}'\mathbf{X}\hat{\beta} = 0$$

da cui:

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

Per le proprietà della forma quadratica minimizzanda, si è certi che la soluzione trovata rappresenta un minimo globale. Formalmente, l'espressione matematica precedente corrisponde ad una proiezione ortogonale delle osservazioni y sullo spazio generato dalle colonne della matrice $\mathbf{X}$. Esisterà pertanto un vettore di pesi γ, tale per cui è possibile ottenere una stima della variabile dipendente come una combinazione lineare delle colonne della matrice dei regressori:

$$\hat{y} = \mathbf{X}\gamma$$

Definiamo anche il seguente modello:

$$y_i - x_i(\beta_i) = \mathbf{X}(\beta_i)\beta_{OLS} + residui \text{ con } i = 1, \ldots, p$$

Questo modello prende il nome di regressione ausiliaria, verrà sfruttato in seguito per il calcolo di alcune misure.

Misure di bontà del fitting: R squared.
In statistica, il coefficiente di determinazione, (più comunemente R^2), è una proporzione tra la variabilità dei dati e la correttezza del modello statistico utilizzato. Esso misura la frazione di varianza della variabile dipendente espressa dalla regressione. Non esiste una definizione concordata di R^2. Nelle regressioni lineari semplici esso è semplicemente il quadrato del coefficiente di correlazione:

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

Dove:

- $ESS = \sum_{i=1}^{n}(\hat{y} - \bar{y})^2$ è la varianza spiegata dal modello (Explained Sum of Squares);
- $TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2$ è la varianza totale (Total Sum of Squares);
- $RSS = \sum_{i=1}^{n}(y_i - \hat{y})^2 = \sum_{i=1}^{n} e_i^2$ è la varianza totale residua (Residual Sum of Squares);
- y_i sono i dati osservati;
- $\bar{y}$ è la media dei dati osservati;
- $\hat{y}$ sono i dati stimati dal modello ottenuto dalla regressione.

R^2 varia tra 0 ed 1: quando è 0 il modello utilizzato non spiega per nulla i dati; quando è 1 il modello spiega perfettamente i dati. Esiste anche un'altra versione di questo coefficiente: l'adjusted R^2 (si indica con $\bar{R}^2$). Mentre R^2 viene utilizzato come principale indice di bontà della curva di regressione per la regressione lineare semplice, $\bar{R}^2$ viene utilizzato per l'analisi di regressione lineare multipla. Esso serve a misurare la frazione di variabilità di Y "spiegata" dalla variabile X. All'aumentare del numero di regressori X, aumenta anche il valore di R^2, per cui spesso $\bar{R}^2$ utilizzato al suo posto. Il coefficiente $\bar{R}^2$ può essere negativo e vale sempre la disuguaglianza $\bar{R}^2 < R^2$.

$$\bar{R}^2 = 1 - (1 - R^2)\frac{n-1}{n-k-1}$$

in cui:

- n è il numero delle osservazioni;
- k è il numero dei regressori.

Se l'$\bar{R}^2$ o l'R^2 sono prossimi ad 1 significa che i regressori predicono bene il valore della variabile dipendente, mentre se è pari a 0 significa che non lo fanno. Questi due coefficienti, tuttavia, non dicono se:

- una variabile sia significativa dal punto di vista statistico;
- i regressori sono causa effettiva dei movimenti della variabile dipendente;
- c'è una distorsione da variabile omessa;
- è stato scelto il gruppo di regressori più appropriato.

Ipotesi sui residui.
Le proprietà degli stimatori dei parametri del modello richiedono alcune assunzioni. È utile verificare la validità di tali assunzioni. Una tecnica di verifica si basa sull'analisi dei residui. Se il modello è ben specificato, i residui rifletteranno le proprietà attribuite ai termini di errore che sono:

- **Normalità**. Gli errori per ogni valore di x hanno una distribuzione normale. Il modello di regressione è robusto rispetto a scostamenti da tale ipotesi: le inferenze su retta e coefficienti di regressione non risultano tuttavia compromesse da una distribuzione degli errori solo approssimativamente normale.
- **Omoschedasticità**. Questa assunzione prevede una variabilità costante per ciascun valore (sia per valori piccoli che elevati, gli errori devono variare di uno stesso ammontare). Tale ipotesi è fondamentale: se non è soddisfatta, occorre trasformare in modo opportuno i dati (per esempio, su scala logaritmica) oppure utilizzare metodi di stima diversa.
- **Indipendenza**. Per ciascun valore dei regressori, questa assunzione assume un ruolo importante quando i dati sono frutto di osservazioni nel corso del tempo. Infatti, gli errori che si producono potrebbero essere correlati ad errori risalenti a periodi temporali precedenti.

Un modo semplice per verificare tali assunzioni è analizzare il grafico dei residui della regressione. Un esempio di tale grafico lo possiamo vedere in figura 4.2.
Una prima assunzione da fare sui residui è quella di linearità. Si assume che la funzione di regressione sia di tipo lineare; nel grafico presente in figura 4.2, è evidente che la relazione è di tipo lineare. Si assume che la varianza delle Y stimate sia costante per ogni valore della variabile esplicativa: in altre parole, occorre che vi sia omoschedasticità. In presenza di omoschedasticità, il grafico dei residui deve presentarsi come una nuvola di punti, come nella figura 4.2; in caso contrario si parla di eteroschedasticità. In terzo luogo, si assume che i dati siano indipendenti:

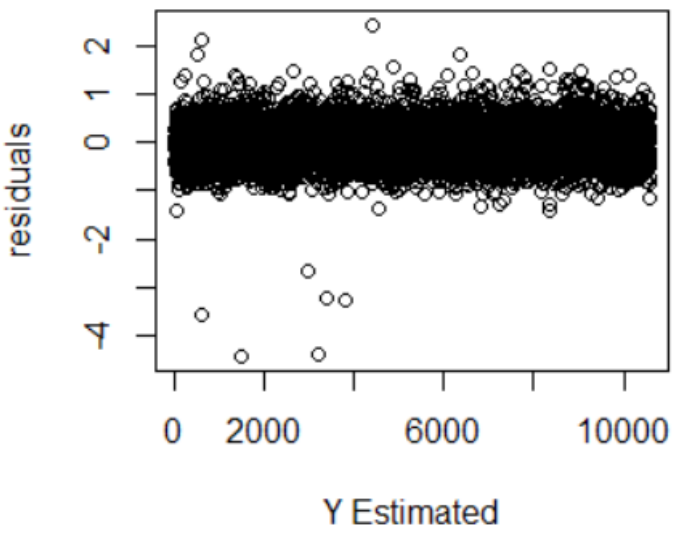


Figura 4.2: Un esempio di grafico dei residui

se le osservazioni fanno parte di una sequenza temporale, in genere gli errori non sono indipendenti. Disponendo in un grafico i residui secondo l'ordine temporale di osservazione, possiamo avere che:

- I residui contigui tendono ad assumere stesso segno, parliamo di autocorrelazione positiva. Sul grafico, i residui mostrano comportamenti ciclici intorno allo zero;
- I residui contigui tendono ad assumere segno opposto: autocorrelazione negativa. Sul grafico i residui tendono a cambiare segno.

L'ultima assunzione è quella di normalità dei dati. Un modo semplice per verificare tale assunzione è considerare i residui standardizzati, che devono distribuirsi al crescere di n, secondo una Normale Standardizzata. Utilizzando il grafico Q-Q plot, come in figura 4.3, possiamo verificare in pochi secondi se i residui si distribuiscono normalmente. Quanto più i punti si allineano lungo la bisettrice, tanto più è verificata l'ipotesi di normalità. Nel grafico in figura 4.3, notiamo che i residui sono ben approssimabili ad una distribuzione normale.

Multicollinearità.
Fra le quattro assunzioni dei minimi quadrati, comprendere il concetto di collinearità è di fondamentale importanza per l'applicazione della Regressione Multipla nella Features Selection. Per questo motivo, vale la pena approfondirlo. La nozione di collinearità deriva dalla geometria vettoriale. Due vettori $\vec{v}$ e $\vec{u}$ si dicono collineari se e solo se esiste uno scalare k tale che sia $\vec{v} = k\vec{u}$ o, in maniera equivalente, $\vec{u} = k\vec{v}$. Collineari, infatti, significa *"giacenti sulla stessa retta"*. Sull'insieme dei vettori non nulli, la relazione di collinearità è:

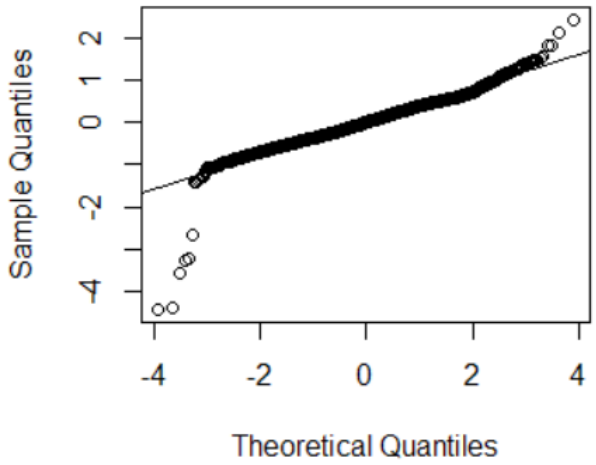


Figura 4.3: Un esempio di Q-Q plot

- Riflessiva: un vettore è collineare con sé stesso;
- Simmetrica: se un vettore $\vec{u}$ è collineare ad un vettore $\vec{v}$, allora $\vec{v}$ è collineare ad un vettore $\vec{u}$;
- Transitiva: se un vettore $\vec{u}$ è collineare ad un vettore $\vec{v}$ e $\vec{v}$ è collineare con $\vec{w}$ allora $\vec{u}$ è collineare con $\vec{w}$.

Queste tre proprietà consentono di affermare che la relazione di collinearità è una relazione di equivalenza. Se due o più colonne della matrice dei regressori $\mathbf{X}$ sono linearmente dipendenti, non esiste l'inversa $(\mathbf{X}'\mathbf{X})^{-1}$, per cui il vettore di stime OLS non può essere determinato. Se è vero che è molto improbabile che ciò si verifichi, nelle applicazioni pratiche può accadere che alcune colonne della matrice dei regressori siano molto vicini ad essere linearmente dipendenti. In questo caso è possibile ottenere un vettore di stime dei minimi quadrati, ma si andrà incontro al problema della multicollinearità. Traslando il concetto dalla geometria al nostro caso, si parla di multicollinearità quando una o più colonne della matrice dei regressori sono prossime ad essere linearmente dipendenti. L'effetto della multicollinearità è che la matrice $\mathbf{X}'\mathbf{X}$ è approssimabile ad una matrice singolare, ovvero una matrice quadrata con determinante zero (ciò implica che non è invertibile). Questo ha due conseguenze importanti:

1. La significatività statistica dei singoli coefficienti è modesta;
2. Il fitting della regressione risulta elevato.

Il primo punto implica che gli intervalli di confidenza per i valori dei coefficienti saranno ampi, quindi le stime dei parametri possono essere molto lontane dai valori reali.

Una conseguenza di un alto grado di multicollinearità è che, anche se la matrice $\mathbf{X}'\mathbf{X}$ è invertibile, un algoritmo potrebbe non riuscire ad ottenere una matrice inversa appropriata ed anche se riuscisse ad ottenerne una potrebbe essere numericamente inaccurata. In presenza di una matrice precisa, si presentano tuttavia le seguenti conseguenze:

- In presenza di multicollinearità, la stima dell'impatto di un regressore sulla variabile dipendente Y tende ad essere meno preciso rispetto al caso in cui i regressori non siano correlati. L' interpretazione di un coefficiente di regressione è che fornisce una stima dell'effetto di una variazione di una unità in una variabile indipendente sulla variabile dipendente, mantenendo costanti gli altri regressori. Se X_1 è altamente correlato con un'altra variabile indipendente X_2, allora abbiamo un set di osservazioni che per X_1 e X_2 hanno una particolare relazione stocastica lineare. Non abbiamo una serie di osservazioni per le quali tutte le modifiche in X_1 sono indipendenti dalle modifiche in X_2, quindi abbiamo una stima imprecisa dell'impatto delle variazioni di X_1 su Y;

- Le variabili collineari contengono le stesse informazioni sulla variabile Y. Anche se sono nominalmente misure "diverse", effettivamente quantificano lo stesso fenomeno. Se le variabili hanno nomi diversi e utilizzano differenti scale di misura numerica ma sono altamente correlate l'una con l'altra, allora soffrono di ridondanza;

- Una delle caratteristiche della multicollinearità è che gli errori standard dei coefficienti interessati tendono ad essere ampi. In tal caso, il test d'ipotesi in base al quale il coefficiente sia uguale a zero, può portare a rifiutare una falsa ipotesi nulla di nessun effetto del regressore, un errore di tipo II;

- Un altro problema con la multicollinearità è che piccole modifiche ai dati di input possono portare a grandi cambiamenti nel modello, con conseguenti cambiamenti nel segno delle stime dei parametri.

- Il principale pericolo di tale ridondanza dei dati è l'overfitting (che ricordiamo essere uno dei motivi della Features Selection) nei modelli di analisi di regressione. I migliori modelli di regressione sono quelli in cui le variabili predittive si correlano ciascuna con la variabile dipendente, ma sono correlate al massimo solo in minima parte l'una con l'altra. Tale modello viene spesso definito "a basso rumore" e sarà statisticamente robusto.

Fintanto che le specifiche sottostanti sono corrette, la multicollinearità in realtà non influenza i risultati, produce solo grandi errori standard nelle relative variabili indipendenti. Ancora più importante, l'obiettivo della regressione è quello di estrarre i coefficienti dal modello ed applicarli ad altri dati. Poiché la multicollinearità causa stime imprecise dei valori dei coefficienti, le previsioni fatte fuori dal campione

saranno imprecise. E se il modello di multicollinearità nei nuovi dati differisce da quello nei dati che sono stati adattati, tale estrapolazione può introdurre errori di grandi dimensioni nelle previsioni.

Multicollinearità a livello generale.
Per individuare la presenza di multicollinearità, possiamo utilizzare delle misure che ci permettono di capire se esistono dei regressori che sono (o sono vicini) alla dipendenza lineare con altri regressori. Queste misure vengono applicate all'intero modello, ciò significa che non sapremo quali sono i regressori linearmente dipendenti fra loro. È tuttavia utile utilizzare queste tecniche preliminarmente per la *multicollinearity detection*, poiché l'analisi a livello di singoli regressori è onerosa. Vediamo di seguito le misure che sono state utilizzate nel nostro caso:

- **Determinante.** Viene calcolato il determinante della matrice $\mathbf{X'X}$. Tale matrice sarà singolare se contiene colonne o righe linearmente dipendenti. Pertanto, il determinante della matrice di correlazione normalizzata senza intercetta ($R = \mathbf{X'X}$), può essere utilizzato per indicare l'esistenza di collinearità tra i regressori. Il determinante della matrice $\mathbf{X'X}$ è compreso tra 0 e 1. Se il determinante è prossimo allo zero si ha collinearità.

- **Farrar Chi-Square.** Viene eseguito un test Chi-quadro per valutare la grandezza della collinearità del set completo di regressori.

$$\chi^2 = -[n - 1 - \frac{1}{6(2p+5)}] \times log_e[\mathbf{X'X}] \sim \psi_{\frac{1}{2}p(p-1)}$$

  La collinearità esiste se $\chi^2 > \chi^2_{\frac{1}{2}p(p-1)}$. In questo caso, indichiamo con p il numero dei regressori, mentre n è il numero dei gradi di libertà, cioè il numero di osservazioni effettuate.

- **Red indicator.** È un indicatore sintetico normalizzato, utilizzato per la multicollinearity detection, che calcola gli autovalori o utilizza la correlazione media dei dati.

$$Red = \frac{\sqrt{\sum_{i=1}^{p}(\lambda_i - 1)^2}}{\frac{p}{\sqrt{p-1}}}$$

  Se tale quantità è vicina allo 0 significa che non c'è ridondanza, se è vicina ad 1 significa che c'è ridondanza nei dati e, di conseguenza, multicollinearità. λ indica l'autovalore, mentre p sono i gradi di libertà.

- **Somma dei lambda inversi.** Un sistema ortogonale $\sum_{i=1}^{p} \frac{1}{\lambda_i} = p$ basato per esempio su una matrice di correlazione $\mathbf{R}$, con autovalori λ_i che confrontano p con la sommatoria presente nella formula precedente, può essere usato per individuare la collinearità. Se il risultato della sommatoria è circa cinque volte più grande del numero dei regressori utilizzati nel modello, allora è presente collinearità.

- **Theil's indicator.** Theil ha proposto una misura di collinearità basata sul contributo incrementale $(R^2 - R_i^2)$ alla correlazione multipla quadrata, dove R_{-i}^2 è l' R^2 della regressione ausiliaria dei regressori.

$$m = R^2 - \sum_{i=1}^{p} R^2 - R_{-i}^2$$

  Se $m = 0$ non esiste ridondanza nei regressori. Al contrario, se $m \sim 1$ esiste ridondanza.

- **Condition number.** La collinearità esiste se questa quantità è superiore a 10, 15 o 30, in base al valore di soglia scelto da chi progetta il modello.

$$CN_i = \sqrt{\frac{max\lambda_i}{\lambda_i}} \text{con } \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p.$$

Individuare la multicollinearità a livello di singolo regressore.
Una volta che è stata individuata la presenza di multicollinearità nel dataset sfruttando le tecniche descritte in precedenza, si può procedere alla ricerca dei regressori collineari, i quali risultano cioè essere linearmente dipendenti fra loro. Anche in questo caso esistono delle misure che consentono di individuare quale regressore è responsabile della ridondanza:

- **VIF.** Misura quanto la varianza di ciascuna stima dei coefficienti di regressione aumenta nel caso in cui non ci sia nessuna correlazione tra i p regressori. Gli elementi che si trovano sulla diagonale della matrice $(\mathbf{X}'\mathbf{X})^{-1}$ sono considerati molto importanti nell'individuazione della multicollinearità.

$$VIF_j = (\mathbf{X}'\mathbf{X})_{jj}^{-1} = \frac{1}{1 - R_j^2}.$$

  Il problema di VIF è che $var(\hat{\beta}_i) = \frac{\sigma^2}{\sum x_i^2 \times VIF_i}$, dipende da σ^2, $\sum x_i^2$ e da VIF, questo vuol dire che un VIF molto elevato può essere controbilanciato

da una bassa σ^2 o da un elevato $\sum x_i^2$. Pertanto, un elevato valore di VIF potrebbe non essere sufficiente per trovare la multicollinearità. È presente multicollinearità se $VIF >$ 3,10,15 questo dipende in base alle scelte di chi progetta il modello.

- **TOL.** È il reciproco di VIF. $TOL = \frac{1}{VIF} = 1 - R_i^2$, valgono le stesse considerazioni che sono già state fatte per VIF, ma al contrario. Una TOL bassa può essere controbilanciata da un'alta σ^2 o da un basso $\sum x_i^2$. È presente multicollinearità se $TOL \sim 0$, anche in questo caso dipende dalle scelte progettuali.

- **W_i.** È un test di Fisher che serve per capire se esiste collinearità tra un regressore e gli altri, calcola una correlazione multipla tra i coefficienti dei regressori.

$$W_i = \frac{R_i^2}{1 - R_i^2} \times \frac{n - p}{p - 1} \sim F_{(n-p,p-1)}$$

  Se $W_i > F_{(n-p,p-1)}$ allora è presente multicollinearità. Ricordiamo che n sono i gradi di libertà, ovvero il numero di osservazioni effettuate dai regressori, mentre p è il numero dei regressori.

- **Metodo di Leamer.** Leamer, nel 2002, suggerisce una misura dell'effetto della multicollinearità per l'i-esima variabile:

$$C_i = \sqrt{\frac{(\sum_{i=1}^{n}(x_{ij} - \bar{x_i})^2)^{-1}}{(\mathbf{X'X})_{ii}^{-1}}}$$

  Questa misura è la radice quadrata del rapporto tra le varianze dei coefficienti stimati, quando la stima è effettuata senza e con gli altri regressori. Se X_i non è correlato con gli altri regressori, C_i sarebbe 1, altrimenti sarebbe uguale a $(1 - R_i^2)^{(1/2)}$, questo significa che $C_i \sim 0$ e che, di conseguenza, esiste collinearità.

- **CVIF.** Questa misura, nata nel 1971, ha come obiettivo quello di valutare l'impatto della correlazione tra i regressori tramite la varianza OLS.

$$CVIF_i = VIF_i \times \frac{1 - R_i^2}{1 - R_0^2} = R_{yX1}^2 + R_{yX2}^2 + \cdots + R_{yi}^2$$

  Esiste collinearità se questo indice è maggiore di 10.

- **Regola di Klein.** Se l'R_i^2 della regressione ausiliaria è maggiore dell'intero R^2 (ottenuto dalla regressione di y su tutti i regressori), allora potrebbe esserci multicollinearità. La regola di decisione per stabilire se è presente multicollinearità è $R^2_{X_j,X_1,X_2,\ldots,X_p} > R^2_{y,X_1,X_2,\ldots,X_p}$.

- **Relazione tra F ed $\mathbf{R^2}$.** La relazione tra la F di Fisher e l'R^2 della regressione di X_i sugli altri regressori rimanenti, può essere utilizzata per rilevare la multicollinearità. La relazione è descritta come:

$$F_i = \frac{R^2_{X_i,X_1,\ldots,X_p}}{p-2} \times \frac{n-p+1}{1-R_{X_i,X_1,\ldots,X_p}}$$

  dove $F^* = F(p-2, n-p-1)$. Se $F_i > F^*$ allora esiste multicollinearità.

La decomposizione di Cholesky della matrice di correlazione

Un altro metodo che è possibile sfruttare per rilevare la multicollinearità è la decomposizione di Cholesky applicata alla matrice di correlazione. Vediamo di seguito di cosa si tratta. Precisiamo che, in questo paragrafo ed in quelli successivi, fra le misure di correlazione esistenti viene utilizzata la correlazione di Pearson. In statistica, l'indice di correlazione di Pearson tra due variabili statistiche, è un indice che esprime un'eventuale relazione di linearità tra di esse. Date due variabili statistiche **X** e **Y**, l'indice di correlazione di Pearson è definito come la loro covarianza, divisa per il prodotto delle deviazioni standard delle due variabili:

$$\rho_{XY} = \frac{\sigma_{XY}}{\sigma_x \sigma_y}$$

Il numeratore è la covarianza tra le due variabili statistiche, mentre il denominatore è dato dal prodotto delle deviazioni standard fra le variabili stesse. Il coefficiente assume sempre valori compresi tra -1 e 1. Nella pratica, possiamo distinguere vari tipi di correlazione:

- Se $\rho_{XY} > 0$, allora le variabili statistiche sono correlate positivamente;

- Se $\rho_{XY} < 0$, allora le variabili statistiche sono correlate negativamente;

- Se $\rho_{XY} = 0$, allora le variabili statistiche non sono correlate.

Per la correlazione positiva (ma analogamente per quella negativa) distinguiamo:

- Correlazione debole, se $0 < \rho_{XY} < 0{,}3$;

- Correlazione moderata, se $0{,}3 < \rho_{XY} < 0{,}7$;

- Correlazione forte, se $\rho_{XY} > 0{,}7$.

Se le due variabili sono indipendenti, allora l'indice di correlazione vale 0 (non vale però il viceversa). La non correlazione, è quindi una condizione necessaria ma non sufficiente per l'indipendenza. L'indice di correlazione vale 1 in presenza di correlazione lineare positiva perfetta, cioè $Y = a + bX, b > 0$; mentre l'indice di correlazione vale -1 in presenza di correlazione lineare negativa perfetta, cioè $Y = a + bX, b < 0$. È possibile che l'indice di correlazione si avvicini a +1 o a -1 anche nel caso in cui le relazioni siano non lineari. Se le variabili statistiche sono più di una, allora è possibile calcolare un indice di correlazione per ciascuna coppia di variabili. Gli indici di correlazione di n variabili possono essere riassunti in una matrice di correlazione, una matrice quadrata di dimensione $n \times n$ avente sia sulle righe che sulle colonne le variabili oggetto di studio. Il generico elemento della matrice ρ_{ij} rappresenta l'indice di correlazione fra la variabile i-esima e la variabile j-esima della matrice. La matrice è simmetrica, cioè $\rho_{ij} = \rho_{ji}$ ed i coefficienti di correlazione sulla diagonale valgono 1, poiché $\rho_{ii} = \frac{\sigma_i i}{\sigma_i i^2}$.
Concentriamoci ora sulla decomposizione di Cholesky. In algebra lineare, la decomposizione di Cholesky è la fattorizzazione di una matrice hermitiana e definita positiva, in una matrice triangolare inferiore e nella sua trasposta coniugata. Ricordiamo che, in algebra lineare:

- Una matrice hermitiana è una matrice complessa costituita che coincide con la trasposta coniugata, pertanto, poiché faremo riferimento ad una matrice hermitiana a valori reali, trattasi di una matrice simmetrica;

- Una matrice definita positiva è una matrice **A** tale che, detto **x*** il trasposto complesso coniugato di **x** (cioè l'elemento generico ottenuto scambiando il suo valore con il complesso coniugato), si verifica che la parte reale di **x*Ax** è positiva per ogni vettore complesso **x** diverso da zero;

- Una matrice triangolare è una matrice quadrata che ha tutti elementi nulli sotto (triangolare superiore) o sopra (triangolare inferiore) la diagonale principale;

- La matrice trasposta coniugata di una generica matrice è ottenuta effettuando la trasposta e scambiando ogni valore con il suo valore complesso coniugato.

Vediamo nel dettaglio il procedimento. Sia **A** una matrice quadrata, hermitiana e definita positiva nel campo dei numeri complessi. Tale matrice può essere scomposta in due matrici: $\mathbf{A} = \mathbf{L}$ x $\mathbf{L}^+$ dove **L** è la matrice triangolare inferiore, con elementi diagonali positivi, mentre $\mathbf{L}^+$ è la matrice coniugata trasposta di **L**. Se la matrice A è reale e simmetrica, la coniugata trasposta di L coincide con la trasposta e la decomposizione si semplifica in $A = L \times L^T$. L'algoritmo di Cholesky, usato per calcolare la matrice di decomposizione L, è una versione modificata dell'algoritmo di Gauss, utilizzato per trasformare una matrice qualsiasi in una matrice a scalini. La matrice L si costruisce procedendo per righe con le formule:

$$L_{j,j} = \sqrt{A_{j,j} - \sum_{k=1}^{j-1} L_{j,k}^2}$$
$$L_{i,j} = \frac{1}{L_{j,j}}(A_{i,j} - \sum_{k=1}^{j-1} L_{i,k}L_{j,k})$$

Applicando la decomposizione di Cholesky alla matrice di correlazione dei regressori, otteniamo un metodo per rilevare la collinearità. Utilizzando come output la trasposta coniugata della matrice triangolare inferiore, otteniamo una matrice triangolare superiore. Esiste collinearità se i valori che si trovano nella diagonale principale sono prossimi allo zero; in caso contrario non è presente collinearità.

P-value e coefficienti di regressione.
Le informazioni del modello di regressione vengono estratte interpretando il significato di due diversi parametri: il p-value ed il coefficiente associati a ciascun regressore. I p-values ed i coefficienti dell'analisi di regressione, ci dicono quali relazioni all'interno del modello sono statisticamente significative. I coefficienti descrivono la relazione matematica tra ciascun regressore e la variabile dipendente. I P-values per i coefficienti indicano se tali relazioni sono statisticamente significative.
Dopo aver valutato il fitting del modello ed esserci assicurati, tramite l'analisi dei residui, di avere delle stime imparziali verificandone le assunzioni, il passo successivo è quello di interpretare l'output statistico della regressione. L'analisi di regressione lineare può produrre molti risultati. Per capire quali relazioni sono rilevanti, dobbiamo essere in grado di interpretare p-values e coefficienti dei regressori.
I p-values ci indicano se le relazioni che osserviamo nel campione utilizzato per la regressione esistono anche in una popolazione più grande. Il p-value per ciascuna variabile indipendente verifica l'ipotesi nulla che la variabile non abbia alcuna correlazione con la variabile dipendente. Se non c'è alcuna correlazione, non vi è alcuna associazione tra i cambiamenti nella variabile indipendente e gli scostamenti della variabile dipendente: non esistono, quindi, evidenze statistiche per affermare che tale effetto esista in generale. Se il p-value per una variabile è inferiore al livello di significatività, i dati del campione forniscono prove sufficienti per respingere l'ipotesi nulla per l'intera popolazione. Se invece i dati supportano l'ipotesi che esista una correlazione diversa da zero, allora le variazioni della variabile indipendente sono associate a cambiamenti nella variabile dipendente nell'intera popolazione. Questo vuol dire che il regressore è statisticamente significativo ed è utile nel modello. Ciò significa che, se la variabile di regressione è significativa per il modello, allora deve essere mantenuta nella Features Selection. D'altra parte, un p-value maggiore del livello di significatività indica che nel campione non ci sono prove sufficienti per arrivare alla conclusione che esiste una correlazione diversa da zero.
Il segno di un coefficiente di regressione indica se esiste una correlazione positiva o negativa tra ciascuna variabile indipendente e la variabile dipendente. Un coefficiente positivo indica che con l'aumentare del valore della variabile indipendente,

anche la media del regressore tende ad aumentare. Un coefficiente negativo, al contrario, indica che con l'aumentare del regressore, la variabile dipendente tende a diminuire. Il valore del coefficiente indica quanto cambia la media della variabile dipendente uno scostamento di un'unità nel regressore mentre si mantengono costanti le altre variabili del modello. Quest'ultimo passaggio è cruciale, perché consente di valutare l'effetto di ciascuna variabile in isolamento rispetto alle altre. Di seguito, nella tabella 4.1 vediamo come interpretare la tabella associata all'output della regressione.

Tabella 4.1: Esempio di un output di una Regressione Lineare

Terms	**Estimate**	**Std.error**	**Statistic**	**p-value**	
(Intercept)	4,978	0,04	13,050	0,000	***
X1	-0,060	0,02	-27,010	0,007	**
X2	-0,045	0,03	-17,240	0,085	.
X3	-0,127	0,04	-29,430	0,003	**
X4	-0,086	0,02	-41,957	0,000	***
X5	-0,127	0,02	-56,771	0,000	***
X6	0,113	0,02	51,888	0,000	***
X7	0,724	0,02	40,743	0,000	***
X8	0,377	0,03	1,416	0,000	***
X9	0,408	0,02	1,795	0,000	***
X11	-0,988	0,05	-1,857	0,000	***
X12	-0,228	0,05	-47,971	0,000	***
X13	-0,055	0,04	-13,116	0,190	
X15	0,042	0,05	0,907	0,364	
X16	0,076	0,07	10,800	0,280	
Residual std error	0,359				
R-squared	0,930				
Adj R-squared	0,929				
Degrees of freedom	10568				

Spieghiamo brevemente il significato delle colonne e delle misure che possiamo notare nella parte bassa della tabella:

- **Estimate.** Sono i coefficienti β_0 e β_1 della regressione. L'informazione sull'intercetta β_0 è data nella prima riga, mentre le informazioni sulla pendenza sono mostrati nella riga che comincia con X_1. Nella nostra tabella, per esempio, $\beta_0 = 4{,}978$ e $\beta_1 = -0{,}06$;
- **Std.Error.** È l'errore standard associato alla stima di β_0 e β_1, nel caso dell'intercetta è calcolato come:

$$\sigma_{\beta_0} = \sigma_E \sqrt{\frac{1}{n} + \frac{x^2}{\sum (x_i - \bar{x})^2}}$$

- **Statistic.** È il valore del test statistico associato alla t di Student con $(n-2)$ gradi di libertà. Un test di Student è un test di ipotesi che verifica l'uguaglianza statistica fra due valori. Nel nostro caso, l'ipotesi nulla è che il valore dello stimatore dell'intercetta (o del regressore) sia uguale al valore reale dell'intercetta stessa (o del regressore stesso);
- **p-value.** Corrisponde alla probabilità di trovare un valore della statistica t maggiore rispetto alla precedente. Se questo valore è inferiore alla soglia di significatività, significa che il regressore rappresenta una feature significativa per il modello e di conseguenza tale feature deve essere aggiunta al sottoinsieme degli attributi rilevanti da mantenere nella nostra analisi dei dati;
- **Residual std error.** Questa è la stima dell'errore della variabilità, σ_E, calcolato come deviazione standard dei residui $s_r = \bar{\sigma_E} = \sqrt{RMS}$. Tale stima è basata su $n - k$ gradi di libertà (10568 nella nostra tabella 4.1);
- **R-squared.** È il coefficiente di determinazione R^2;
- **Adj R-squared.** È il coefficiente di determinazione aggiustato $\bar{R^2}$;
- **Degrees of freedom.** Sono i gradi di libertà, dati dalla differenza $n - k$ in cui n rappresentano le osservazioni e k i regressori.

ANOVA e test di Fisher.
Un metodo alternativo alla regressione multipla lineare è rappresentato dal test-F utilizzato dall'ANOVA. ANOVA (ANalysis Of VAriance) sfrutta dei test di Fisher per valutare l'uguaglianza statistica delle medie quando nel modello sono coinvolte tre o più variabili. Come sappiamo, il test di Fisher utilizza l'omonima distribuzione per effettuare un test di ipotesi: la statistica F è il rapporto fra due varianze. É difficile dare un'interpretazione diretta alle varianze, poiché esse sono espresse in unità quadratiche rispetto ai dati di partenza. La deviazione standard, invece, essendo espressa nella stessa unità dei dati, è più semplice da interpretare come misura di dispersione. Per questo motivo, è preferibile utilizzare la deviazione standard come misura di dispersione; la varianza, tuttavia, è utilizzata da alcuni test statistici. Una statistica F è il rapporto fra due varianze, o tecnicamente, due varianze campionarie. Le varianze campionarie sono stimatori delle varianze (la varianza, infatti, è stimata perché la calcoliamo sul campione).
Dato che i test di Fisher valutano il rapporto tra due varianze, si potrebbe pensare che essi siano adatti solo per determinare se le varianze dei due campioni siano

uguali o no. In realtà l'F-test può essere utile per molti altri scopi, poiché esso è molto flessibile. Per esempio, gli F-test possono testare l'importanza complessiva nei modelli di regressione o determinare se un set di medie sono uguali da un punto di vista statistico. Il test di Fisher di significatività globale indica se il modello di regressione lineare fornisce un adattamento migliore ai dati rispetto ad un modello che non contiene variabili indipendenti. Prendiamo in considerazione come il test di Fisher di significatività complessiva si adatta alle altre statistiche di regressione come l'R^2: quest'ultimo ci indica quanto bene il modello di regressione si adatta ai dati, mentre l'F-test indica quanto il modello è correlato ad essi. Il test di Fisher confronta il modello di regressione con lo stesso modello senza la variabile indipendente; un test di questo tipo prende il nome di *intercept-only model*. L'F-test ha le seguenti ipotesi:

- L'ipotesi nulla afferma che il modello senza variabili indipendenti si adatta ai dati ed al modello;

- L'ipotesi alternativa dice che il modello si adatta meglio ai dati rispetto all'intercept-only model.

L'obiettivo è trovare un set di medie che siano uguali. Per valutare questo con un F-test, occorre utilizzare le opportune varianze nel rapporto. Riportiamo di seguito il rapporto utilizzato dalla statistica F per l'ANOVA ad una via.

$$F = \frac{Varianza_{gruppi}}{Varianza_{gruppo}}$$

I risultati dell'ANOVA vengono riassunti in una tabella che mostra lo split della varianza totale nelle variabili coinvolte nella regressione e nei residui. Nella tabella 4.2 riassumiamo l'output dell'ANOVA. Spieghiamo ora come interpretarlo, esattamente come abbiamo fatto in precedenza per la regressione lineare.

- **Term.** In questa colonna sono elencati i regressori ed i residui;

- **Sum.sq.** È la quantità di varianza spiegata dalla regressione lineare, è conosciuta anche con il nome di Regression Sum of Squares o ancora come Model Sum of Squares. È data dalla formula seguente:

$$Regression_{SS} = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

- **Mean.sq.** La regressione o model mean squares (MS) è uguale alla Regression Sum of Squares, divisa per i gradi di libertà. Nel caso della regressione, poiché i gradi di libertà sono pari ad 1, Sum sq e Mean sq si equivalgono nei regressori;

- **Statistic.** È una statistica F calcolata per valutare se è possibile spiegare una quantità significativa di variabilità dalla regressione lineare sulla variabile dipendente. Ogni regressore ha il suo F-value. Esso è calcolato come:

$$F = \frac{MS_{regression}}{MS_{residual}}$$

  La statistica F segue una distribuzione di Fisher con $(k-1, n-k) = (1, n-2)$ gradi di libertà. L'F-value può assumere solo valori positivi, valori elevati di questa quantità stanno a significare che la regressione lineare spiega una quantità significativa di variabilità

- **P-value.** È la probabilità di osservare una statistica F più grande rispetto a quella osservata nel punto precedente. Nel nostro caso, per esempio, il p-value riferito alla variabile X_1 è molto prossimo allo 0 (Excel non riesce a distinguerlo da 0), ciò sta ad indicare che esiste un'evidenza molto forte di una relazione lineare tra la variabile dipendente Y ed il regressore X_1.

- **Response.** Indica qual è la variabile dipendente, nel nostro esempio essa si chiama semplicemente Y;

Tabella 4.2: Esempio di un output di ANOVA

Term	**Df**	**Sumsq**	**Meansq**	**Statistic**	**P-value**	
X1	1	293,45	293,45	197,68	0,000	***
X2	1	134,07	134,07	90,31	0,000	***
X3	1	245,87	245,87	165,63	0,000	***
X4	1	572,57	572,57	385,71	0,000	***
X5	1	143,18	143,18	96,45	0,000	***
X6	1	6184,07	6184,07	4165,89	0,000	***
X7	1	264,54	264,54	178,20	0,000	***
X8	1	1043,38	1043,38	702,87	0,000	***
X9	1	132,37	132,37	891,70	0,000	***
X11	1	506,15	506,15	340,97	0,000	***
X12	1	370,07	370,07	249,29	0,000	***
X13	1	0,25	0,25	1,68	0,194	
X15	1	0,11	0,11	0,76	0,384	
X16	1	0,17	0,17	1,17	0,280	
Residuals	8817	130,88	0,15	NA	NA	

4.2.2 DBSCAN.

Il *DBSCAN* (Density Based Spatial Clustering of Application with Noise) è il primo algoritmo di clustering che ha introdotto il concetto di densità. In precedenza, abbiamo considerato algoritmi di cluster basati sulla distanza. Gli algoritmi di clustering che utilizzano il concetto di densità, individuano i cluster come regioni dello spazio caratterizzati da un'elevata densità di pattern, separate dalle regioni meno dense. Un vantaggio di questo algoritmo è l'individuazione di cluster di forma arbitraria, ma soprattutto il DBSCAN consente di identificare gli *outliers*. In virtù di tali considerazioni, gli algoritmi di cluster *density-based* consentono di migliorare la performance del risultato finale. Lo scopo del DBSCAN è quello di includere in un unico cluster gli elementi che si trovano nella stessa regione dello spazio caratterizzata da un'alta densità di pattern.
Poiché anche nel DBSCAN è presente un concetto di distanza, si potrebbe erroneamente ipotizzare che i pattern siano considerati come vettori, in modo da creare uno spazio vettoriale. In realtà, il DBSCAN è una tecnica applicabile a qualsiasi tipo di pattern, non solo a pattern vettoriali.
Il DBSCAN è caratterizzato dai seguenti parametri:

- **eps**: rappresenta il raggio massimo tale per cui un pattern è considerato *density-reachable* da un altro;
- **minPts**: rappresenta il numero minimo di punti che devono essere situati all'interno di un raggio eps.
- **Eps-neighborhood di un pattern p**: indicando la distanza euclidea tra due pattern p e q con $d(p, q)$ definiamo $N_{eps}(p)$ come il numero di pattern che si trovano entro un raggio ϵ da p.

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

  dove D rappresenta il database che comprende tutti gli elementi.

Écosì possibile individuare quali punti fanno parte di un cluster, verificando se il valore di *Eps-neighborhood* contiene un numero minimo di punti *minPts*.
L'approccio dell'*Eps-neighborhood* è troppo semplicistico, in quanto occorre distinguere i pattern inclusi nel cluster definiti *core point*, da quelli che sono situati ai margini del cluster che sono chiamati *border point*. Tale misura è difficile da controllare, perché se ϵ è troppo grande allora tutti i pattern vengono inclusi in un unico cluster, mentre se ϵ è troppo piccolo DBSCAN non riesce a fornire nessun risultato. É proprio per questo che si ricorre al parametro *minPts*, che misura il numero di punti contenuti nel cluster in $N_\epsilon(p)$. Tale definizione è debole nel considerare la differenza tra due pattern differenti all'interno del cluster, perché deve raccogliere il numero maggiore di punti in maniera da raggruppare tutti i pattern all'interno

dello stesso cluster, definendo per i punti periferici un valore minimo possibile in modo da distinguerli dal rumore. Definiamo pertanto:
Core point: un punto p è definito core point o punto interno al cluster se nel suo intorno di raggio ϵ sono presenti almeno *minPts* punti: $\|N_\epsilon(q)\| \geq minPts$.
Consideriamo due punti p e q. Il punto p si dice raggiungibile (secondo il concetto di raggiungibilità diretta) dal punto q, relativamente ai parametri ϵ e *minPts*, se $p \in N_{epsilon}(q)$; $N_{epsilon}(q) \geq minPts$, ovvero se q è un core point. La raggiungibilità diretta è simmetrica per una coppia di pattern definiti come core point, mentre non vale fra un border point ed un core point (un border point è un punto interno al cluster che però non soddisfa la condizione di core point).
Il punto p è raggiungibile dal punto q se esiste una catena di punti $p_1, \ldots, p_n$ con $p_1 = q$ e $p_n = p$, tale che p_{i+1} sia raggiungibile da p_i. Questa relazione non presenta la proprietà di simmetria. Infatti il punto q potrebbe essere un border point e potrebbe avere un numero insufficiente di vicini per essere considerato un punto "genuino" del cluster.
Il punto p si dice connesso al punto q, in relazione ai parametri ϵ e *minPts* se esiste un punto o, tale che p e q siano raggiungibili da o in relazione agli stessi parametri ϵ e *minPts*. Tale relazione è riflessiva e simmetrica. Nella figura 4.4 è rappresentata la differenza fra le definizioni di raggiungibilità e connessione.

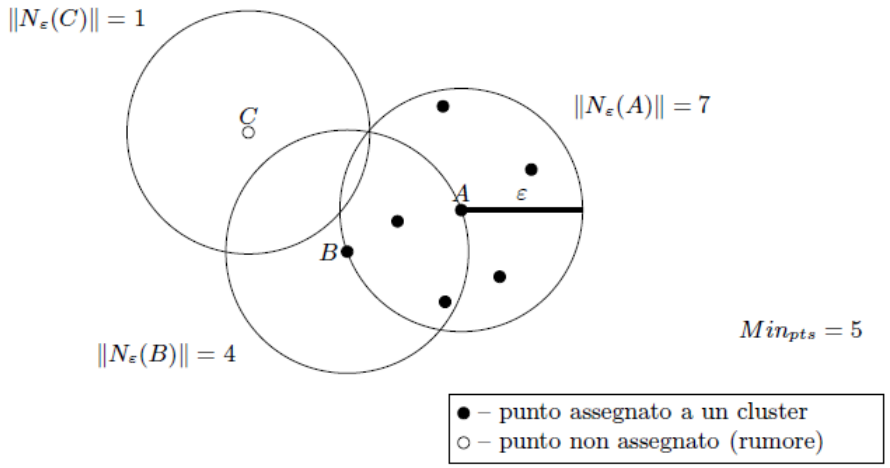


Figura 4.4: Esempio di output del DBSCAN. A è un *core point*, B un *border point* e C un *outlier*

Fatte queste doverose premesse, possiamo ora definire i concetti principali su cui si basa questo algoritmo.
Dato un database D ed i parametri *minPts* ed ϵ, un cluster C è l'insieme non vuoto in cui sono valide le seguenti proprietà:

1. $\forall p, q$ se $p \in C$ e q è raggiungibile dal punto p (core point), allora $q \in C$ secondo il concetto di raggiungibilità diretta;

2. $\forall p, q \in C$, p è connesso a q secondo il concetto di connettività introdotto precedentemente.

Un cluster, secondo un algoritmo *density-based*, è un insieme in cui risulta massimale la connettività sui punti. É da questo concetto che gli outliers sono definiti come pattern che non appartengono ad alcun cluster.
Dati i cluster $C_1, \ldots, C_k$ appartenenti al database D, generati in base ai parametri *minPts* e ϵ, il rumore è l'insieme dei punti di D che non appartiene a nessun cluster: $outliers = \{p \in D \mid \forall i : p \notin C_i\}$.
Prima di mostrare lo pseudocodice dell'algoritmo DBSCAN, enunciamo i seguenti teoremi:

- Sia p un punto di D e $\|N_{epsilon}(p)\| \geq minPts$, allora l'insieme $O = \{o \mid o \in D \wedge o\}$ è raggiungibile da p, relativamente ai parametri ϵ e *minPts* è un cluster in relazione ai parametri ϵ e *minPts*.

- Sia C un cluster in relazione ai parametri ϵ e *minPts* e p un punto di C tale che $\|N_{\epsilon}(p)\| \geq minPts$, allora C coincide con l'insieme $O = \{o \mid o \in D \wedge o\}$ è raggiungibile da p, relativamente ai parametri ϵ e *minPts* .

Di seguito, riportiamo lo pseudocodice del DBSCAN.

Algoritmo DBSCAN. ©Susi, Dulli, Furini

input: D insieme dei punti
ϵ
N_{min}
output: partizione in cluster di D

```
1: CId=next(Id);
2: for each p ∈ D do
3:     if p non è stato assegnato a nessun cluster then
4:         if expand (D,p, CId, ϵ, N_min) then
5:             CId=next(Id);
6:         end if
7:     end if
8: end for
```

Algoritmo DBSCAN: expand. ©Susi, Dulli, Furini

input: D insieme dei punti
ϵ
N_{min}
p
N_{min}Cid

output: boolean

```
1:  seeds=D.query(p, ε);
2:  if ||nseeds|| < N_min then
3:      p.C=Noise;
4:      return false;
5:  end if
6:  for each q ∈ seeds do
7:      q.C=CId;
8:  end for
9:  seeds.Remove(p);
10: while ||seeds|| > 0 do
11:     p=seeds.RemoveHead();
12:     R=D.query(p, ε);
13:     if ||R|| ≥ N_min then
14:         while ||R > 0 do
15:             q=R.RemoveHead();
16:             if q.C== then
17:                 eeds.Append(q);
18:                 q.C=CId;
19:             end if
20:             if q.C==Noise then
21:                 q.C=CId;
22:             end if
23:         end while
24:     end if
25: end while
26: return true;
```

Calcolo dei parametri del DBSCAN.
Come possiamo intuire, per progettare un DBSCAN che abbia una performance elevata, occorre comprendere come settare i parametri ϵ e *minPts*. Questa tecnica, a differenza degli algoritmi partizionati, non necessita della conoscenza a priori del numero di cluster, nè dei pattern iniziali dai quali estendere la ricerca. Richiede tuttavia che siano predeterminati i parametri definiti precedentemente.
Per una corretta individuazione degli outliers, è fondamentale seguire un metodo rigoroso e preciso con cui calcolare ϵ e *minPts*. Esistono in letteratura diversi metodi per poter settare i parametri del DBSCAN. La tecnica utilizzata in questa tesi è rappresentata dal *K nearest neighbors (KNN)*.
Scelto il numero K, è possibile definire la funzione $K - distance$ che assegna a ciascun pattern del database D la distanza dal suo k-esimo *nearest neighbors*. L'idea sottostante è che all'interno del cluster, ci aspettiamo che i K *nearest neighbors* stiano a circa la stessa distanza. Un outlier avrà il K *nearest neighbors* ad una distanza maggiore rispetto ai core points ed ai border points. In seguito, i pattern del database D vengono disposti in ordine decrescente, con lo scopo di ottenere il grafico di figura 4.5 chiamato *sorted k-distance plot.*

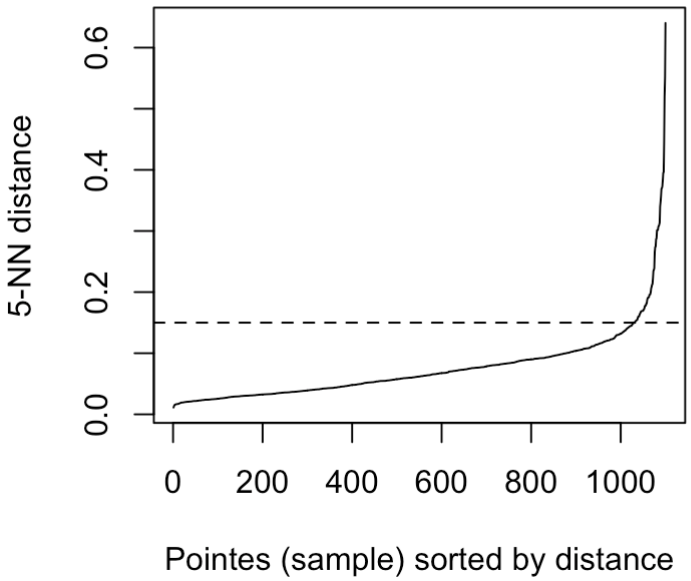


Figura 4.5: Esempio di sorted k-distance plot

Nel grafico, il valore di ϵ viene estratto dall'ordinata di p, mentre il valore di $minPts$ è rappresentato da K, che nel nostro esempio vale 5.

Valutazione dell'algoritmo. Il risultato finale del DBSCAN dipende in larga misura dalla scelta del valore K, pertanto se il valore di K è corretto, l'andamento della curva rimane similare. La critica maggiore che viene sollevata contro il DBSCAN è proprio il metodo utilizzato per calcolare i parametri ϵ e $minPts$: questi devono necessariamente essere calcolati per tentativi, facendo variare K fin quando non si nota che l'andamento del *KNN dist plot* rimane quasi costante. Spesso la ricerca di tali parametri è un processo che richiede tempi di calcolo piuttosto lunghi. La complessità computazionale del DBSCAN è infatti pari a $O(n)$, a cui va sommata la complessità della query che deve essere eseguita, nel caso peggiore, per ogni punto. Il DBSCAN non funziona correttamente se:

- i cluster hanno una densità variabile: in questo caso i parametri ϵ e $minPts$ variano a seconda dei cluster, ma l'algoritmo non permette di definire coppie di parametri differenti per ciascun cluster;

- la misura della distanza soffre in caso di *high-dimensional data*. Ricordiamo che il DBSCAN sfrutta la distanza euclidea, un tipo di distanza che comporta una difficoltosa ricerca del valore appropriato di ϵ.

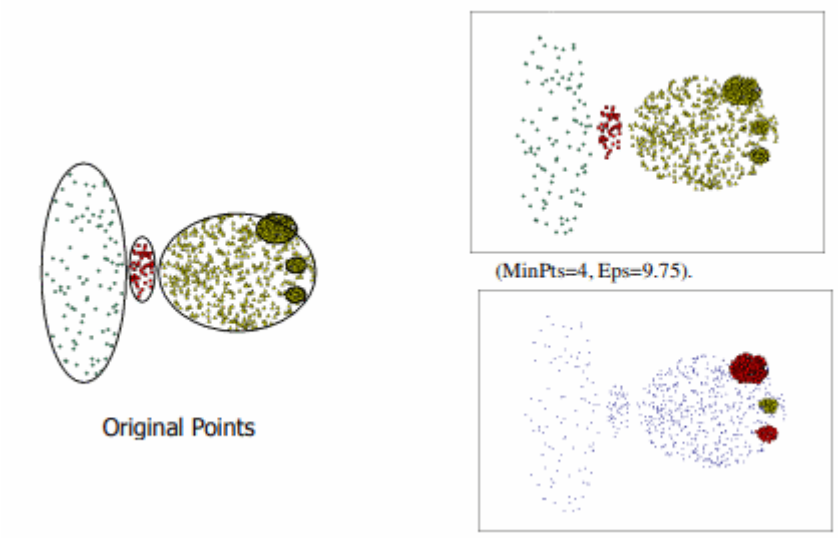


Figura 4.6: Esempio di clustering mal riuscito col DBSCAN. ©Tan, Steinbach, Kumar

Ciononostante, il DBSCAN costituisce una tecnica rigorosa ed ottima per l'individuazione degli outliers. La letteratura lo esalta, inoltre, per la sua capacità di riconoscere cluster di forme arbitrarie e non necessariamente convesse, come possiamo notare nella figura 4.7

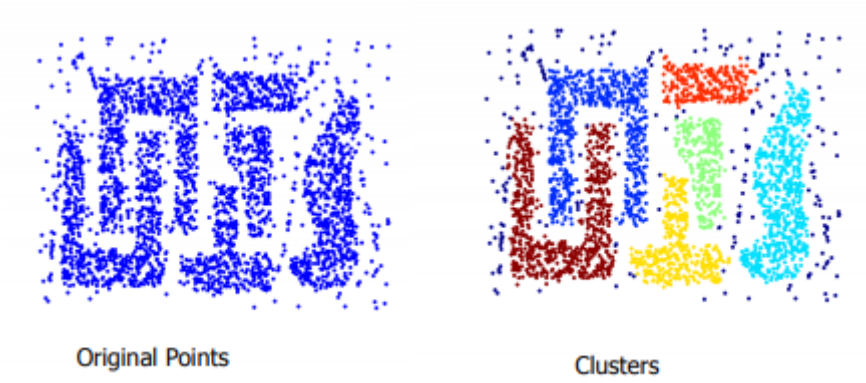


Figura 4.7: Esempio di clustering ben riuscito con il DBSCAN. ©Tan, Steinbach, Kumar

4.3 Processo di analisi ed estrazione della conoscenza

La fase dell'estrazione della conoscenza nascosta viene implementata da F-SCAN utilizzando un algoritmo di clustering partizionato: il *K-means*. Questo effettua una partizione dei dati con l'obiettivo di inserire all'interno dello stesso cluster gli edifici che possiedono caratteristiche termo-fisiche simili. Prima di effettuare il clustering sul dataset, quest'ultimo è stato normalizzato. Nella sezione successiva, descriviamo nel dettaglio il K-means.

4.3.1 K-means

L'algoritmo *K-means* appartiene alla famiglia degli algoritmi partizionativi, che consentono di suddividere il dataset in K partizioni in base ai loro attributi.
Il K-means è un algoritmo progettato nel 1967, che si pone come obiettivo la minimizzazione della varianza *inter-cluster*. Ogni cluster viene identificato con il suo centroide (punto medio). L'algoritmo segue una procedura iterativa: inizialmente crea K partizioni ed assegna a ciascuna di esse i punti d'ingresso casualmente, oppure utilizzando alcune informazioni euristiche. Successivamente, calcola il centroide di ogni gruppo e costruisce una nuova partizione, associando ogni oggetto in ingresso al cluster il cui centroide è più vicino ad esso. Dopo che un punto è stato assegnato al cluster, vengono ricalcolati i centroidi. La procedura viene iterata fin quando l'algoritmo non converge.

Descrizione formale.
Dati N oggetti caratterizzati da i attributi, questi possono essere modellati come vettori contenuti in uno spazio vettoriale con i dimensioni. Definiamo:

$$X = \{X_1, X_2, \ldots, X_N\}$$

l'insieme degli oggetti. Si definisce partizione degli oggetti il seguente gruppo di insiemi:

$$P = \{P_1, P_2, \ldots, P_K\}$$

che possiedono le proprietà seguenti:

- Tutti gli oggetti devono appartenere almeno ad un cluster: $\bigcup_{i=1}^{K} P_i = X$
- Ogni oggetto può appartenere ad un solo cluster: $\bigcap_{i=1}^{K} P_i = \emptyset$
- Almeno un oggetto deve appartenere ad un cluster e nessun cluster può contenere tutti gli oggetti: $\emptyset \subset A_i \subset X$

Inoltre, una proprietà ovvia che deve essere soddisfatta è che il numero delle partizioni deve essere inferiore al numero degli oggetti presenti nel dataset ($1 \leq K \leq N$): così come risulta essere inutile avere un unico cluster in cui sono presenti tutti gli oggetti, è altrettanto inutile avere tanti cluster composti esclusivamente da un unico oggetto. Rappresentiamo le partizioni mediante delle matrici $U \in \mathbb{N}^{K \times N}$, dove l'elemento generico $u_{i,j} \in \{0,1\}$, indica se l'oggetto i appartiene al cluster j. Inoltre, l'insieme $C = \{C_1, C_2 \ldots, C_K\}$ è l'insieme dei centroidi. Passate in rassegna queste definizioni, possiamo ora definire la funzione obiettivo del K-means:

$$V(U, C) = \sum_{i=1}^{K} \sum_{X_j \in P_i} \|X_j - C_i\|^2$$

L'obiettivo dell'algoritmo è quello di minimizzare tale funzione, applicando la procedura operativa che è stata spiegata in precedenza:

1. Generazione U_v e C_v casuali;
2. Calcolo U_n che minimizzi la funzione obiettivo $V(U, C_v)$;
3. Calcolo C_n che minimizzi la funzione obiettivo $V(U_v, C)$;
4. Verifica della conversione: in caso affermativo si ferma, altrimenti si torna al passo 2 ponendo $U_v = U_n$ e $C_v = C_n$.

Il $K - means$ riconosce la convergenza se non c'è stato alcun cambiamento nella matrice U, oppure se la differenza fra i valori che la funzione obiettivo $V(U, C)$ assume in due iterazioni successive non supera una soglia predeterminata. Di seguito, riportiamo nel dettaglio i passi dell'algoritmo K-means.

Valutazione dell'algoritmo.

Algoritmo K-means. ©Susi, Dulli, Furini.

input: insieme di N pattern $\{x_i\}$
numero di cluster desiderato K
output: insieme di K cluster

1: Setta il numero K di pattern da utilizzare come centri $\{c_j\}$ dei cluster;
2: **repeat**
3: Assegna ciascun pattern x_i al cluster P_j che abbia il centro c_j più vicino a x_i
4: Ricalcola i centri $\{c_j\}$ dei cluster utilizzando i pattern che appartengono a ciascun cluster, utilizzando la media o il centroide;
5: **until** *criterio di convergenza soddisfatto*

L'algoritmo ha riscosso successo perché riesce ad ottenere la convergenza in tempi relativamente brevi. In generale, il numero di iterazioni che vengono eseguite prima di incontrare la convergenza è minore rispetto al numero dei punti. É stato dimostrato che esistono alcuni insiemi di punti per cui l'algoritmo impiega un tempo pari a $2^{\Omega(\sqrt{n})}$ per convergere, ossia un tempo superpolinomiale. Se da un lato le prestazioni in termini di tempo computazionale sono ottime, dall'altro l'algoritmo non assicura il raggiungimento della soluzione ottima globale. La soluzione finale è influenzata da alcuni fattori, specialmente dal set di cluster iniziale e nella pratica è possibile ottenere un output che si discosta molto dall'ottimo globale. Tuttavia, è possibile sfruttare l'elevata rapidità del K-means per poterlo applicare più volte e scegliere fra le soluzioni prodotte quella maggiormente soddisfacente. Uno svantaggio non indifferente è che l'algoritmo richiede come input il numero K di cluster con cui effettuare la partizione del dataset e funziona bene solamente se i cluster hanno forma sferica. La misura di valutazione più comune per la bontà di questa tecnica di clustering è l'*SSE (Sum of Squared Error)*. Per ogni punto x, l'SSE è dato dalla distanza che intercorre fra il punto x ed il centroide C_i del cluster A_i a cui il punto stesso viene assegnato:

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

Si dimostra che il centroide che minimizza tale misura di valutazione, è la media dei punti del cluster:

$$C_i = \sum_{x \in A_i} x$$

Precisiamo che quest'ultima affermazione è vera solamente se la misura di distanza utilizzata è quella euclidea.
Come si può dedurre dalla formula dell'SSE, questa misura si riduce all'aumentare di K, ma questo non significa che il numero ideale di cluster con cui partizionare il dataset debba essere un numero alto. Un clustering ottimo con un K ridotto può avere un valore di SSE minore rispetto ad un pessimo clustering con un K elevato. Quando il K-means riassegna i punti ai centroidi lo fa in base alle distanze minori, pertanto il computo dei nuovi centroidi minimizza il valore della misura di valutazione per il cluster, ma talvolta capita che la scelta dei centroidi iniziali possa portare a delle soluzioni non accettabili, come possiamo vedere in figura 4.8.
Nella figura 4.8, possiamo notare che nella soluzione in basso a destra, spostare un centroide comporta sempre un aumento dell'SSE, tuttavia è preferibile la soluzione che troviamo in basso a sinistra.
Come abbiamo accennato, i centroidi iniziali spesso sono scelti casualmente dall'algoritmo K-means, alcune volte essi si riposizioneranno in maniera corretta, altre volte no. Esistono tuttavia alcune soluzioni per superare il problema della selezione dei centroidi iniziali:

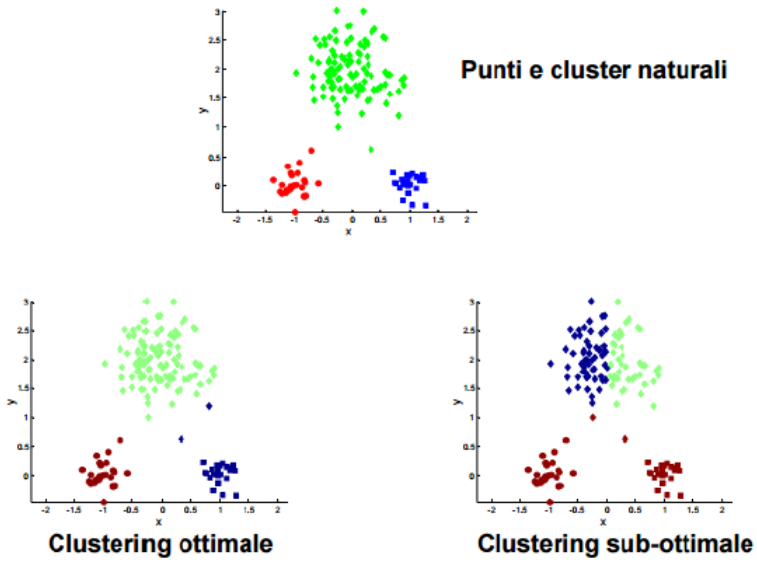


Figura 4.8: K-means: i centroidi iniziali portano a soluzioni non accettabili. ©Tan, Steinbach, Kumar

- lanciare il K-means diverse volte, ciascuna delle quali inizializzandola con centroidi di partenza diversi;
- effettuare un campionamento dei punti per poi eseguire una tecnica di clustering gerarchico;
- scegliere un numero K più elevato di quello reale per poi ridurlo aggregando i cluster simili, ovvero quelli che condividono alcuni attributi, mantenendo i cluster con una maggiore distanza inter-cluster;
- effettuare il post-processing tramite tecniche che servono per eliminare i cluster che sono stati individuati per errore.

É possibile che il K-means possa portare alla presenza di cluster vuoti, causata dal fatto che durante l'assegnamento ad un centroide non venga assegnato nessun record. Ciò può portare ad un SSE molto alto perché un cluster non viene utilizzato. Esistono diversi metodi per individuare un opportuno centroide da utilizzare in alternativa, tipicamente ciò viene eseguito selezionando un centroide che porta alla divisione di un cluster in due differenti, che includono gli elementi più vicini.
L'algoritmo K-means produce scarsi risultati quando:

- i cluster hanno diverse dimensioni, in quanto se non sono ben separati l'SSE porta al settaggio di centroidi che individuano cluster delle stesse dimensioni;

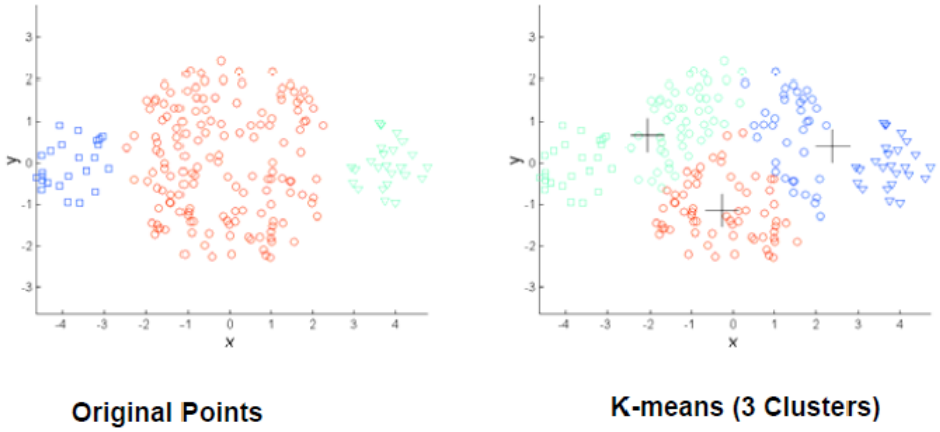


Figura 4.9: Cluster di diverse dimensioni portano il K-means ad una soluzione non valida. ©Tan, Steinbach, Kumar

- i cluster hanno una diversa densità, poiché questo determina delle distanze all'interno del cluster minore, il che significa che le zone a densità minore richiedono più punti mediani per minimizzare l'SSE;

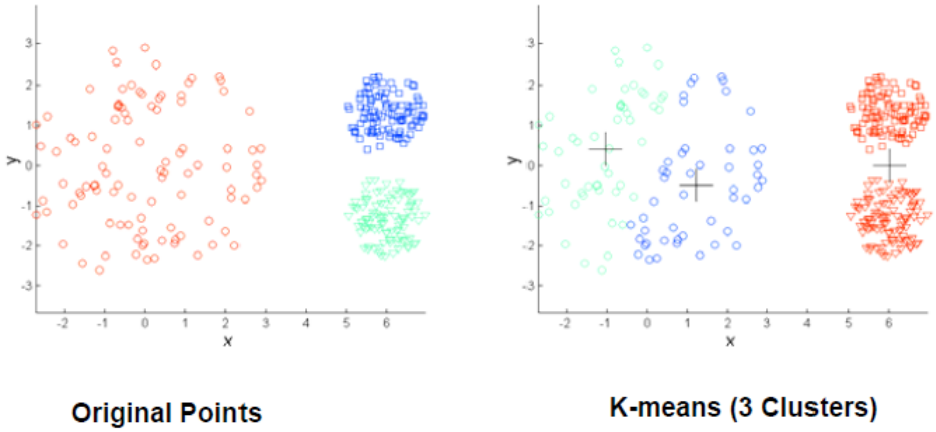


Figura 4.10: Cluster di diversa densità portano il K-means ad una soluzione non valida. ©Tan, Steinbach, Kumar

- i cluster hanno forme globulari, perché l'SSE è calcolato in base alla distanza euclidea che non considera la forma degli oggetti;

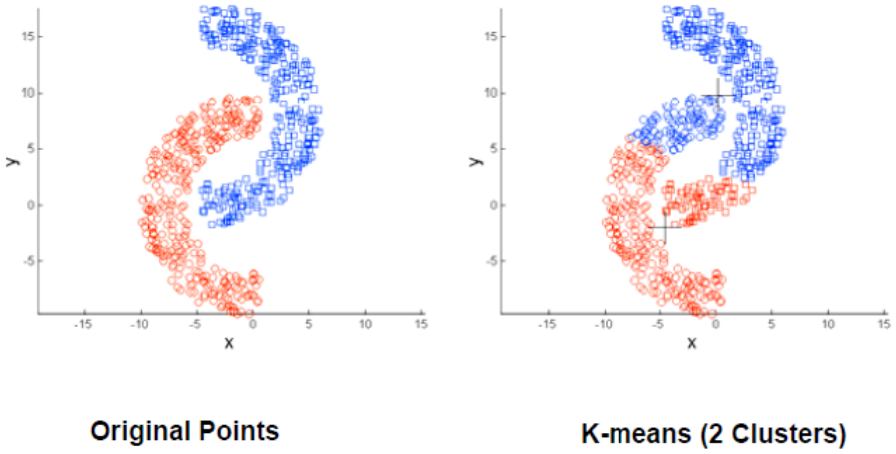


Figura 4.11: Cluster globulari portano il K-means ad una soluzione non valida. ©Tan, Steinbach, Kumar

- sono presenti outliers, poiché il rumore tende a modificare il centroide del cluster ed i punti molto lontani incidono sul suo valore. Tuttavia, nella presente tesi tale problema non si pone, poiché viene applicato il DBSCAN per eliminare i dati rumorosi.

Scelta del numero K. La performance del K-means dipende in larga misura dal settaggio di K: questo costituisce un input del K-means e rappresenta la quantità di cluster con cui l'algoritmo stesso partizionerà l'insieme di *items*. Esistono diversi metodi utili per calcolare il valore ottimale di K. Riportiamo di seguito quelli che sono stati utilizzati nella presente tesi:

- **Elbow method.** L'*Elbow method* è un metodo progettato nel 1953 da Robert Thorndike, utile per trovare il numero appropriato di cluster in un dataset. Tale metodo considera la percentuale della varianza totale presente nel dataset, spiegata come una funzione del numero di cluster: dovremmo scegliere il numero di cluster in modo che l'aggiunta di un ulteriore cluster non fornisca una migliore modellazione dei dati. Più precisamente, se si calcola la percentuale di varianza spiegata dai cluster rispetto al numero di cluster, i primi porteranno un'aggiunta di informazioni enorme, spiegheranno cioè molta della varianza totale; successivamente il guadagno marginale diminuirà, formando appunto un gomito (*elbow*) nel grafico *Numero cluster-percentuale varianza*. Verrà scelto il K che si trova in corrispondenza del gomito. La percentuale della varianza spiegata è il rapporto tra la varianza fra i gruppi e la varianza totale.

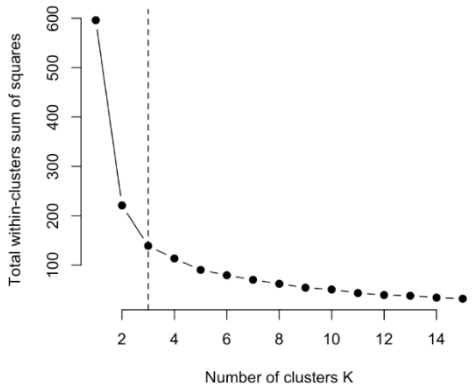


Figura 4.12: Esempio di grafico *Numero cluster-percentuale varianza*: il gomito si forma in corrispondenza di K=3

- **Silhouette.** La Silhouette è una tecnica che fornisce una rappresentazione grafica sintetica di quanto ogni oggetto si trova correttamente all'interno del suo cluster. Di fatto, la Silhouette è una misura che indica quanto un oggetto è maggiormente simile al proprio cluster rispetto agli altri. Varia in un intervallo compreso tra $\{-1, +1\}$, in cui un valore alto di Silhouette indica che l'oggetto è molto simile a quelli contenuti all'interno del suo cluster e scarsamente simile ai cluster vicini. Se la maggior parte degli oggetti ha un valore di Silhouette elevato, la configurazione del clustering è appropriata. Se al contrario molti punti hanno un valore basso o negativo, la configurazione del clustering non è appropriata. La Silhouette può essere calcolata con qualsiasi metrica di distanza. Per ogni oggetto i indichiamo con $a(i)$ la dissimilarità media di i con tutti gli altri oggetti all'interno dello stesso cluster (minore è il valore, migliore è l'assegnazione); $b(i)$ è la minore dissimilarità media di i verso qualsiasi altro cluster, di cui i è non è membro. La Silhouette è misurata come:

$$s(i) = \frac{b(i) - a(i)}{max\left\{a(i), b(i)\right\}}$$

$$s(i) = \begin{cases} 1 - a(i)/b(i) & a(i) < b(i) \\ 0 & a(i) = b(i) \\ b(i)/a(i) - 1 & a(i) > b(i) \end{cases}$$

La media $s(i)$ su tutti i dati del dataset misura quanto i dati sono stati clusterizzati in maniera appropriata e quanto sono simili gli oggetti presenti

nello stesso cluster. In conclusione, la scelta ricade sul K che ha un valore di Silhouette alto.

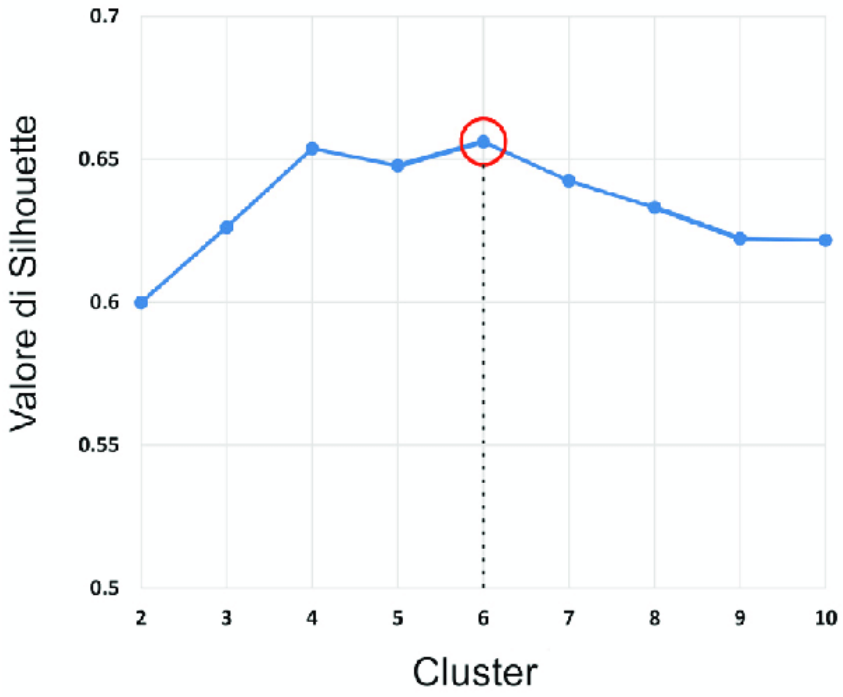


Figura 4.13: Scelta di K con Silhouette. In questo caso, scegliamo $K = 6$

Ora indichiamo con:

- n le osservazioni effettuate;
- p le variabili del dataset;
- q i cluster;
- n_k gli oggetti contenuti nel cluster C_k;
- c_k il centroide del cluster C_k;
- x_i il vettore con p-dimensionale delle osservazioni dell'i-esimo oggetto nel cluster C_k;
- $W_q = \sum_{k=1}^{q} \sum_{i \in C_k} (x_i - c_k)(x_i - c_k)^T$ è la matrice di dispersione intra-gruppo per i dati clusterizzati in q cluster.

Definiamo quindi questi ulteriori indici, utilizzati anch'essi per la scelta del K ottimale:

- **Hartigan index.** L'indice Hartigan è stato proposto nel 1975, si calcola con la seguente equazione.

$$Hartigan = (\frac{trace(W_q)}{trace(W_{q+1})} - 1)(n - q - 1)$$

dove $q \in \{1, \ldots, n-2\}$. La massima differenza tra i livelli gerarchici del clustering viene presa come un'indicazione del numero corretto di cluster nel dataset

- **Scott index.** Indice introdotto da Scott e Symons nel 1971, calcolato come:

$$Scott = n \log \frac{det(T)}{det(W_q)}$$

dove n è il numero di elementi presenti nel dataset, T è la *total sum of squares* e W_q è la somma dei quadrati all'interno del cluster q. La massima differenza tra i livelli gerarchici del clustering è utilizzata per suggerire il valore corretto di K per partizionare il dataset.

- **Trcovw index.** Questo indice, proposto da Milligan e Cooper nel 1985, rappresenta la *trace* all'interno dei cluster all'interno della matrice *pool*

$$Trcovw = trace(Cov(W_q))$$

La differenza massima tra i punteggi dei differenti livelli è utilizzata per indicare la soluzione ottimale

- **KL index.** L'indice KL è stato proposto nel 1988 da Krzanowski e Lai, ed è definito dalla seguente equazione:

$$KL(q) = \left| \frac{DIFF_q}{DIFF_{q+1}} \right|$$

dove $DIFF_q = (q-1)^{2/p} trace(W_{q-1}) - q^{2/p} trace(W_q)$. Il valore di q che massimizza l'indice KL è utilizzato come numero ottimale di cluster con cui partizionare il dataset.

- **Sdbw index.** Questo indice è basato sui criteri di coerenza e separazione tra cluster. Esso è calcolato utilizzando la seguente equazione: $SDbw(q) = Scat(q) + Density.bw(q)$. Il primo termine, $Scat(q)$ è definito come:

$$Scat(q) = \frac{1/q \sum_{k=1}^{q} \|\sigma^{(k)}\|}{\|\sigma\|}$$

dove:

- σ è il vettore delle varianze, per ogni variabile presente nel dataset: $\sigma = ((Var(V_1), Var(V_2), \ldots, Var(V_p)))$
- $\sigma^{(k)}$è il vettore delle varianze per ogni cluster C_k: $\sigma^{(k)} = (Var(V_1^{(k)}, Var(V_2^{(k)}, \ldots, Var(V_p^{(k)}))$

Il secondo termine $Dis(q)$ è calcolato utilizzando l'equazione riportata di seguito, indica la separazione totale tra tutti i q cluster, è un indice della distanza inter-cluster:

$$Dis(q) = \frac{D_{max}}{D_{min}} \sum_{k=1}^{q} (\sum_{z=1}^{q} \|c_k - c_z\|)^{-1}$$

dove

- $D_{max} = max(\|c_k - c_z\|) \forall k, z \in \{1,2,3, \ldots, q\}$ è la massima distanza fra i centri dei cluster;
- $D_{min} = min(\|c_k - c_z\|) \forall k, z \in \{1,2,3, \ldots, q\}$ è la minima distanza tra i centri dei cluster.

Il numero di cluster q che minimizza questo indice, viene considerato un valore ottimale di K con cui partizionare il dataset.

Una volta che è stato scelto il numero K di cluster con cui partizionare il dataset, viene effettuata l'analisi della partizione utilizzando gli strumenti grafici seguenti:

- (i) decomposizione a valori singolari, che riduce la dimensionalità del dataset in modo che la partizione possa essere visualizzata in uno spazio tridimensionale;
- (ii) box-plot (o diagrammi a scatole e baffi), servono per visualizzare come le variabili utilizzate si distribuiscono all'interno dei cluster, verificando se l'algoritmo di clustering è riuscito a separare bene gli edifici con performance energetiche diverse.

Decomposizione a valori singolari.
In algebra lineare, la decomposizione a valori singolari o *Singular Value Decomposition (SVD)* è un metodo di fattorizzazione di una matrice basato sull'uso di autovettori e autovalori. Data una matrice M a valori reali o complessi, avente dimensione $m \times n$, l'SVD è una scrittura del tipo:

$$M = U \sum V^*$$

dove U è una matrice unitaria[1] avente dimensioni $m \times m$, $\sum$ è una matrice diagonale rettangolare di dimensioni $m \times n$ e V^* è la matrice trasposta coniugata di una matrice unitaria V di dimensioni $n \times n$. Gli elementi appartenenti alla matrice $\sum$ sono detti valori singolari della matrice M; ciascuna delle m colonne della matrice U è chiamata *vettore singolare sinistro* mentre ognuna delle n colonne della matrice V è detta *vettore singolare destro*. Si dimostra che i vettori singolari di sinistra di M sono gli autovettori di MM^*; i vettori singolari di destra di M sono gli autovettori di M^*M; i valori singolari non nulli di M (che si trovano nella diagonale principale di $\sum$) sono le radici quadrate degli autovalori non nulli di MM^* e M^*M. Traslando questi concetti dall'algebra lineare all'analisi dei dati, l'obiettivo è quello di ridurre il rango del dataset in modo da generalizzare alcune proprietà. L'SVD cerca di ridurre una matrice avente rango r in una matrice di rango t, solamente se è possibile considerare un insieme di vettori r linearmente indipendenti ed approssimarli con t vettori linearmente indipendenti.

I box-plot.

I box-plot sono una rappresentazione grafica utile per individuare raggruppamenti di attributi numerici utilizzando i loro quartili. Nonostante la loro semplicità, questi strumenti grafici riescono a riassumere molte informazioni utilizzando pochi numeri. Spesso il box-plot è utilizzato per individuare graficamente la presenza di outliers nel dataset, senza ricorrere al DBSCAN. Nella presente tesi, invece, sono stati usati per analizzare la caratterizzazione di un cluster.
Un box-plot, come possiamo notare in figura 4.14, è composto da:

- una mediana, che indica il valore centrale della distribuzione;

- i quartili, forniscono un'indicazione sulla variabilità utilizzando lo scarto interquartile, calcolato come la differenza tra il terzo ed il primo quartile: $W = Q_3 - Q_1$. W indica l'intervallo entro cui ricade il 50% dei valori;

- i valori estremi, ovvero il valore maggiore ed il valore minore della distribuzione, indicano l'intero intervallo della distribuzione ma anche la presenza di outliers, rappresentati da singoli punti.

La posizione della mediana rispetto ai quartili, indica anche il grado di simmetria della distribuzione.

[1] una matrice unitaria è una matrice quadrata complessa A che soddisfa la condizione $AA^T = A^TA = I$.

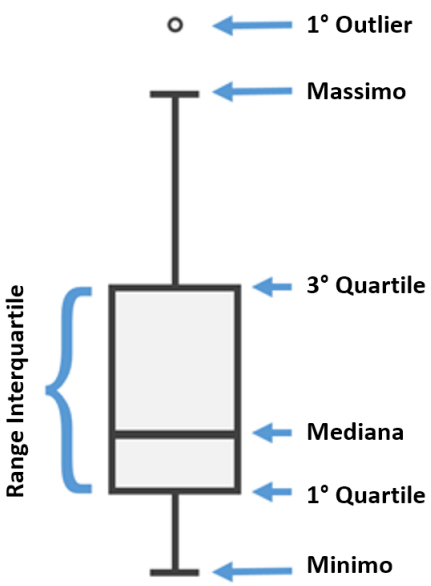


Figura 4.14: Esempio di boxplot

4.3.2 Classificatore ad albero.

Una volta che è stato effettuato il clustering e che gli edifici simili sono stati raggruppati in base ai loro attributi, si esegue la *cross-validation* tramite la costruzione di un classificatore ad albero. Abbiamo già spiegato nel paragrafo 2.3.2, il funzionamento del decision-tree. In particolare, abbiamo visto che l'algoritmo sottostante è l'ID3. Ricordiamo che tale algoritmo tratta solamente valori discreti, il nostro dataset è composto da attributi numerici continui, trattandosi di dati che rappresentano caratteristiche termo-fisiche. Per questo motivo, il classificatore ad albero che è stato utilizzato per la presente tesi non implementa l'algoritmo ID3, ma utilizza una sua estensione: l'algoritmo C4.5. Rispetto all'algoritmo precedente, C4.5 presenta i seguenti miglioramenti:

- Manipolazione di attributi continui e discreti: l'algoritmo determina un valore di soglia per ciascun attributo scelto come *decision attribute* e poi divide gli oggetti in quelli il cui valore assunto da tale attributo è al di sopra della soglia e quelli che sono pari o inferiori ad esso;

- Manipolazione di dati mancanti: l'attributo con i valori mancanti non viene utilizzato nel calcolo del guadagno e dell'entropia;

- Manipolazione di attributi con costi differenti;

- Potatura albero dopo che è stato generato: l'algoritmo risale dalle foglie alla radice e rimuove i rami che non sono utili per la comprensione del classificatore.

Valutazione dei metodi di classificazione: matrici di confusione. Passiamo in rassegna i metodi di valutazione che servono per analizzare la bontà di un modello di classificazione. Per valutare il modello di classificazione utilizzato nella presente tesi, si è deciso di ricorrere alle matrici di confusione. É necessario precisare che nei modelli di previsione, i risultati effettivi sono tipicamente peggiori delle previsioni, per questo poi è necessario rimettere mano alla modellazione per poter avere un certo tipo di risposta. Un classificatore può essere rappresentato come una funzione che mappa gli elementi di un dataset in classi o gruppi. Se la classificazione avviene con supervisione, come nel nostro caso, l'insieme dei dati che devono essere classificati sono già stati suddivisi in classi: il classificatore serve per valutare la qualità del risultato prodotto. Per introdurre le matrici di confusione e le misure da esse utilizzate, consideriamo per semplicità un classificatore binario, in cui il dataset è suddiviso in due classi che indichiamo convenzionalmente come positiva (p) e negativa n. Gli esiti di un classificatore di questo tipo rientrano in una delle seguenti categorie:

1. Il classificatore produce il valore p' partendo da un dato appartenente alla classe p. Diciamo in questo caso che il classificatore ha prodotto un valore vero positivo (VP);

2. Il classificatore produce il valore p' partendo da un dato appartenente alla classe n. Diciamo in questo caso che il classificatore ha prodotto un valore falso positivo (FP);

3. Il classificatore produce il valore n' partendo da un dato appartenente alla classe n. Diciamo in questo caso che il classificatore ha prodotto un valore vero negativo (VN);

4. Il classificatore produce il valore n' partendo da un dato appartenente alla classe p. Diciamo in questo caso che il classificatore ha prodotto un valore falso negativo (FN).

Dato un set di istanze ed un classificatore, la matrice 2×2 che si forma è detta matrice di confusione, la possiamo notare in tabella 4.3. Esistono diverse misure per valutare le performances di un classificatore, riportiamo di seguito le più frequenti:

$$TFP = \frac{FP}{VN + FP};$$

$$TVP = \frac{VP}{FN + VP};$$

Tabella 4.3: Matrice di confusione

	Classi previste	
Classi effettive	p	n
p'	Veri positivi (VP)	Falsi positivi (FP)
n'	Falsi negativi (FN)	Veri negativi (VN)

$$precisione = \frac{VP}{VP + FP};$$

$$recall = \frac{VP}{VP + FN};$$

$$accuratezza = \frac{VP + VN}{VP + FP + VN + FN}.$$

Nella matrice di confusione, è possibile visualizzare nella diagonale gli items che sono stati interpretati in maniera corretta; gli altri sono gli errori sulle varie classi. In particolare, è possibile determinare due importanti misure di validità di un test: *sensibilità* e *specificità*. La sensibilità è data dal rapporto tra il numero di veri positivi al totale delle istanze positive:

$$sensibilità = \frac{VP}{VP + FN}$$

La specificità invece è dal rapporto tra i veri negativi ed il totale delle istanze negative:

$$specificità = \frac{VN}{VN + FP}.$$

Abbiamo preso come esempio un classificatore binario, in modo da definire in maniera semplice la matrice di confusione e le misure ad essa associate per poterne valutare la performance. La matrice di confusione di un classificatore ad m classi sarà una matrice $m \times m$, in cui il generico elemento $a_{i,j}$ indica il numero di tuple della classe i che sono state etichettate dal classificatore come classe j. In generale, affinché un classificatore abbia una buona precisione, la maggior parte delle tuple dovrebbero comparire lungo la diagonale della matrice di confusione. Idealmente, vorremmo che dalla voce $a_{1,1}$ alla voce $a_{m,m}$, con il resto degli elementi della matrice pari o quasi vicini allo zero. In questo modo, avremmo FP ed FN intorno al valore nullo che alzano il valore delle misure di performance.

4.4 Visualizzazione della conoscenza

La conoscenza estratta viene visualizzata e sintetizzata mediante l'utilizzo di differenti strumenti grafici e tabellari:

- grafici a torta;
- grafici a barre;
- box-plot;
- grafici a dispersione;
- tabelle riepilogative.

Capitolo 5

Risultati sperimentali

L'applicazione degli strumenti descritti nei capitoli precedenti è stata effettuata su dati sperimentali in collaborazione con alcuni esperti di dominio, i quali hanno avuto un ruolo di primaria importanza nella fase di selezione degli attributi utili e della determinazione dei domini di ciascuna caratteristica; inoltre essi sono stati fondamentali anche nel validare i risultati ottenuti.
Le analisi sono state condotte utilizzando il software RapidMiner e l'ambiente di sviluppo RStudio (per maggiori dettagli si rimanda alla lettura del capitolo 3) su un pc con sistema operativo Windows 10, che possiede un processore i5-2467M avente una CPU a 1.60GHz ed una RAM di 4,00 GB.
Il dataset su cui è stato applicato il framework F-SCAN è un estratto del Catasto delle certificazioni energetiche della Regione Piemonte contenente i certificati emessi nel primo semestre del 2013. Nella tabella 5.1 vengono riassunti gli attributi che sono stati selezionati in base agli obiettivi della nostra analisi. Rimandiamo il lettore al capitolo 1, nel quale è presente una loro descrizione dettagliata.
Il Catasto delle certificazioni energetiche della Regione Piemonte raccoglie i certificati relativi a tutti i settori immobiliari. Tuttavia, la nostra analisi è stata svolta considerando solamente gli edifici che hanno una destinazione d'uso E.1,E.1(1), cioè gli edifici residenziali. Lo studio è in una fase iniziale, pertanto abbiamo ritenuto opportuno tralasciare gli edifici aventi destinazione d'uso differente, poiché per questi è più complicato individuare regole di dominio che tengano conto delle loro caratteristiche specifiche. Inoltre, nel catasto regionale, la maggioranza delle certificazioni energetiche si riferiscono ad unità abitative.
Gli scopi principali dell'F-SCAN, applicato alle certificazioni energetiche, sono:

- includere nell'analisi solamente gli attributi significativi per il clustering;
- suddividere le certificazioni in modo da formare cluster simili dal punto di vista della performance energetica;
- fornire all'utente una chiara visualizzazione dei risultati.

Tabella 5.1: Riassunto degli attributi utilizzati nell'analisi

Caratteristiche termo-fisiche	Volume lordo riscaldato (V)
	Superficie disperdente totale (S)
	Superficie utile (S_u)
	Fattore Forma (S/V)
	Trasmittanze opache (U_{op}
	Trasmittanze trasparenti (U_w)
Rendimenti	Rendimento di generazione decimale (η_{gn})
	Rendimento di distribuzione decimale (η_d)
	Rendimento di regolazione decimale (η_e)
	Rendimento di emissione decimale (η_{rg})
	Rendimento globale riscaldamento Torino ($\eta_{g,To}$)
	Rendimento globale acqua calda sanitaria ($\eta_{g,W}$)
	Rendimento globale riscaldamento e acqua calda sanitaria ($\eta_{g,R,W}$)
	Rendimento stagionale acqua calda sanitaria Torino ($\eta_{s,W,To}$)
	Rendimento medio globale stagionale acqua calda sanitaria ($\eta_{g,s,W}$)
Indici di fabbisogno	Fabbisogno energia termica utile (Q_h)
	Fabbisogno energia termica utile
	acqua calda sanitaria ($Q_{h,W}$)
	Fabbisogno energia termica utile Torino ($Q_{h,To}$)
	Fabbisogno acqua calda sanitaria soddisfatto
	da fonti rinnovabili ($Q_{W,FR}$)
Indici di prestazione	Indice di prestazione energetica
	acqua calda sanitaria Torino ($EP_{i,W,To}$)
	Indice di prestazione riscaldamento Torino ($EP_{i,To}$)
	Indice di prestazione energetica globale Torino ($EP_{L,To}$)
	Indice di prestazione energetica acqua calda sanitaria
	fonti rinnovabili Torino ($EP_{i,W,To,FR}$)
	Prestazione energetica acqua calda sanitaria check ($EP_{i,W,check}$)
	Potenza riscaldamento (W)
	Prestazione raggiungibile (EP_L*)
	Classe energetica

Il blocco *Selezione ed integrazione dei dati*, il primo dell'F-SCAN[1], ha portato alla selezione degli attributi presenti nella tabella 5.1. Il passo successivo, è quello di applicare il secondo blocco dell'F-SCAN, ovvero il *Processo di ottimizzazione*. Tuttavia, prima di descrivere tale blocco, occorre fare delle premesse.

[1]Si rimanda il lettore al capitolo 4 per capire come è strutturato il framework e quali sono i concetti teorici sottostanti ad ogni blocco

Le informazioni relative alle certificazioni energetiche estratte dal Catasto e registrate durante il primo semestre 2013, sono state suddivise in due dataset differenti: un dataset D_1, composto da 16 attributi e già sottoposto ad una procedura di *Data Cleaning* nel corso di un altro lavoro di tesi, come abbiamo citato nei capitoli precedenti; un dataset D_2, composto da 28 attributi selezionati con l'aiuto degli esperti di dominio, coerentemente con gli obiettivi che la nostra analisi vuole raggiungere. Di seguito, elenchiamo gli attributi facenti parte del dataset D_1:

- Superficie utile;
- Altezza media;
- Fattore forma;
- Trasmittanze opache;
- Trasmittanze trasparenti;
- Rendimento di generazione decimale;
- Rendimento di regolazione decimale
- Rendimento di distribuzione decimale;
- Rendimento di emissione decimale;
- Rendimento globale riscaldamento Torino;
- Fabbisogno energia termica utile Torino;
- Indice di prestazione energetica acqua calda sanitaria Torino;
- Potenza riscaldamento;
- Rendimento medio globale stagionale acqua calda sanitaria;
- Indice di prestazione energetica acqua calda sanitaria riscaldata da fonti rinnovabili;
- Rendimento stagionale acqua calda sanitaria Torino.

In questo secondo elenco, sono elencati gli attributi inclusi nel dataset D_2:

- Superficie utile;
- Volume lordo riscaldato;
- Altezza media;

- Superficie disperdente totale;
- Fattore forma;
- Trasmittanze opache;
- Trasmittanze trasparenti;
- Fabbisogno energia termica utile;
- Fabbisogno energia termica utile acqua calda sanitaria;
- Rendimento acqua calda sanitaria;
- Prestazione energetica acqua calda sanitaria check;
- Rendimento di generazione decimale;
- Rendimento di regolazione decimale;
- Rendimento di distribuzione decimale;
- Rendimento di emissione decimale;
- Rendimento medio globale check;
- Rendimento globale riscaldamento Torino;
- Rendimento globale riscaldamento e acqua calda sanitaria Torino;
- Fabbisogno di energia termica utile Torino;
- Indice di prestazione riscaldamento Torino;
- Indice di prestazione energetica acqua calda sanitaria Torino;
- Indice di prestazione energetica globale Torino;
- Potenza riscaldamento;
- Fabbisogno di acqua calda sanitaria soddisfatto da fonti rinnovabili;
- Rendimento medio globale stagionale acqua calda sanitaria;
- Prestazione raggiungibile;
- Classe energetica.

Ognuno di questi due dataset è stato trattato dal framework F-SCAN. Tuttavia, gli esperti di dominio suggeriscono che le performances energetiche degli edifici sono influenzate soprattutto da quattro attributi: fattore forma, rendimento medio globale di riscaldamento e produzione di acqua calda sanitaria corretto con i gradi giorno, trasmittanza delle superfici opache, trasmittanza delle superfici trasparenti. L'F-SCAN applica i suoi blocchi sui dataset D_1 e D_2 considerando tutti gli attributi, ma l'analisi dei singoli cluster viene svolta solamente prendendo in considerazione i quattro attributi suggeriti dagli esperti di dominio.
Preventivamente al *Processo di ottimizzazione* è stata applicata la normalizzazione, in modo che le variabili con piccole ampiezze non vengano considerate meno significative rispetto alle variabili caratterizzate da ampiezza maggiore. A titolo di esempio, il rendimento medio globale presenta valori compresi nell'intervallo [0.4,1] mentre le trasmittanze spaziano in intervalli più ampi: [0.15,1.1] le opache mentre quelle trasparenti possono assumere valori fino a 5.5.
La scelta è stata quella di non utilizzare soltanto la normalizzazione Z-score, che è la più applicata quando si esegue il clustering: si è optato anche per la normalizzazione min-max scegliendo l'intervallo [0,1]. Tale scelta non è casuale, infatti la trasformazione mantiene la positività dei valori evitando di snaturare il problema, in quanto i dati trattati sono caratteristiche termo-fisiche e di conseguenza non possono assumere valori minori di 0. Abbiamo a che fare con quattro dataset differenti, avendo applicato due normalizzazioni diverse a D_1 e D_2.
Successivamente, si procede con il blocco del *Processo di ottimizzazione*, costituito dalla Feature Selection implementata tramite Regressione Multipla Lineare ed ANOVA, e dal DBSCAN. Si è pensato a lungo in quale ordine attuare le due tecniche di ottimizzazione del dataset; alla fine, è stato deciso di seguire entrambe le strade: per ogni normalizzazione, abbiamo applicato prima il DBSCAN e poi la Feature Selection da un lato; dall'altro abbiamo invertito l'ordine di esecuzione delle due tecniche in modo da metterci in condizione di poter scegliere quale delle due alternative fornisce il risultato migliore.
In seguito, una volta eliminati gli outliers e gli attributi poco significativi, il dataset può passare al blocco del *Processo di analisi ed estrazione della conoscenza*, che utilizza il K-Means per eseguire il clustering con l'obiettivo di raggruppare i palazzi omogenei da un punto di vista della prestazione energetica. Poiché questo algoritmo di clustering partizionativo utilizza come input il numero K di cluster con cui esso deve suddividere il dataset, abbiamo usufruito dell'ausilio di alcune tecniche che ci consentono di calcolare il valore ottimale di K. Per la scelta del numero K di cluster con cui partizioneremo il dataset, abbiamo pensato di utilizzare il *majority model*, cioè di scegliere il K suggerito dalla maggioranza degli indici utilizzati. Ma dal momento che spesso le soluzioni fornite dagli indici erano discordanti, abbiamo svolto prove differenti per valori di K diversi.
Per ciascun esperimento attuato con un dato valore di K, abbiamo analizzato come l'algoritmo K-means ha eseguito la partizione attraverso utilizzando i box-plot,

che consentono di visualizzare la caratterizzazione di ciascun cluster, e attraverso l'uso di diagrammi a dispersione delle componenti ottenute dalla *Singular Value Decomposition (SVD)*. Grazie all'ausilio di questi due strumenti grafici, abbiamo individuato quale valore di K partiziona meglio il dataset in analisi, così che la migliore partizione venga studiata con la *cross-validation* tramite il classificatore ad albero. Il classificatore ad albero, rispetto alle altre tecniche di *cross-validation*, consente una migliore leggibilità delle regole che è possibile estrarre.
Il passaggio successivo è quello di reperire dal dataset iniziale, estraibile dal catasto delle certificazioni energetiche della Regione Piemonte, altri attributi utili per l'*Esplorazione della conoscenza estratta* ed infine riassumere i risultati finali con grafici di sintesi.
Poiché sono state percorse numerose strade prima di trovare la tecnica migliore con cui effettuare il *Processo di ottimizzazione* ed il *Processo di estrazione della conoscenza*, riassumiamo tutti gli esperimenti effettuati nella tabella 5.2, le cui colonne riportano le seguenti informazioni:

- IdEsp: è il codice identificativo dell'esperimento effettuato;
- Dataset: corrisponde al dataset su cui è stato effettuato l'esperimento. D1 si riferisce al dataset con 16 attributi iniziali e già soggetto a *Data Cleaning* in un lavoro di tesi precedente; D2 invece è il dataset con 28 attributi;
- Normalizzazione: indica la normalizzazione utilizzata nel corso dell'esperimento;
- Applicazione: indica l'ordine di applicazione delle tecniche utilizzate. Indichiamo con:
  - A1: DBSCAN e K-Means;
  - A2: DBSCAN, Regressione e K-means;
  - A3: Regressione, DBSCAN e K-means.
- Attributi: indica il numero di attributi significativi individuati dal processo di ottimizzazione;
- K: indica il valore di K utilizzato come input del K-means in quell'esperimento specifico.

Le righe evidenziate in grassetto nella tabella corrispondono agli esperimenti in cui il K utilizzato, a parità di condizioni, partiziona meglio il dataset di partenza. Come possiamo notare, sono stati effettuati 33 esperimenti in totale. Per ovvie ragioni non possiamo procedere con l'analisi dei risultati di ciascun esperimento. Pertanto abbiamo evidenziato nella tabella i due esperimenti che analizzeremo nella presente

IdEsp	Dataset	Normalizzazione	Applicazione	Attributi	K
E1	D1	min-max	A1	16	3
E2	D1	min-max	A1	16	4
E3	D1	min-max	A1	16	5
E4	D1	min-max	A1	16	6
E5	**D1**	**min-max**	**A1**	**16**	**7**
E6	D1	min-max	A1	16	8
E7	D1	Z-score	A1	16	3
E8	D1	Z-score	A1	16	4
E9	D1	Z-score	A1	16	8
E10	**D1**	**Z-score**	**A1**	**16**	**9**
E11	D1	min-max	A2	11	4
E12	**D1**	**min-max**	**A2**	**11**	**8**
E13	D1	min-max	A2	11	9
E14	D1	Z-score	A2	9	4
E15	D1	Z-score	A2	9	8
E16	D1	Z-score	A2	9	10
E17	D1	min-max	A3	14	4
E18	**D1**	**min-max**	**A3**	**11**	**7**
E19	D1	min-max	A3	14	8
E20	D1	min-max	A3	14	9
E21	D1	Z-score	A3	10	4
E22	**D1**	**Z-score**	**A3**	**10**	**6**
E23	D1	Z-score	A3	10	7
E24	D2	min-max	A3	14	3
E25	D2	min-max	A3	14	9
E26	D2	Z-score	A3	12	4
E27	D2	Z-score	A3	12	6
E28	**D2**	**Z-score**	**A3**	**12**	**8**
E29	**D2**	**min-max**	**A2**	**10**	**6**
E30	D2	min-max	A2	10	9
E31	D2	Z-score	A2	14	5
E32	**D2**	**Z-score**	**A2**	**14**	**8**
E33	D2	Z-score	A2	14	9

Tabella 5.2: Tabella riassuntiva degli esperimenti effettuati

tesi: questi sono gli esperimenti $E5$ ed $E18$. Abbiamo scelto proprio questi perché sono facilmente confrontabili: hanno la stessa normalizzazione e lo stesso valore di K come input del K-means. Con l'analisi di tali risultati sperimentali, dimostreremo che l'applicazione A3 porta a conclusioni migliori rispetto all'applicazione A1, la

quale non prevede il processo di ottimizzazione da noi implementato. L'esperimento $E18$ fornisce i migliori risultati fra tutti quelli i tentativi che abbiamo effettuato.

5.1 Processo di ottimizzazione: Feature Selection e DBSCAN

Avendo già trattato il processo di selezione dei dati nell'introduzione del presente capitolo, possiamo procedere con l'analisi della Feature Selection tramite Regressione Lineare Multipla ed ANOVA. Questo processo è stato interamente implementato utilizzando le tecnologie messe a disposizione dall'ambiente di implementazione RStudio (si rimanda il lettore al Capitolo 3 per maggiori dettagli sui pacchetti impiegati). Come detto in precedenza, gli esperti di dominio suggeriscono di utilizzare i quattro attributi più significativi per caratterizzare ciascun cluster risultante. In figura 5.1 vediamo come questi attributi si distribuiscono nel dataset di partenza.

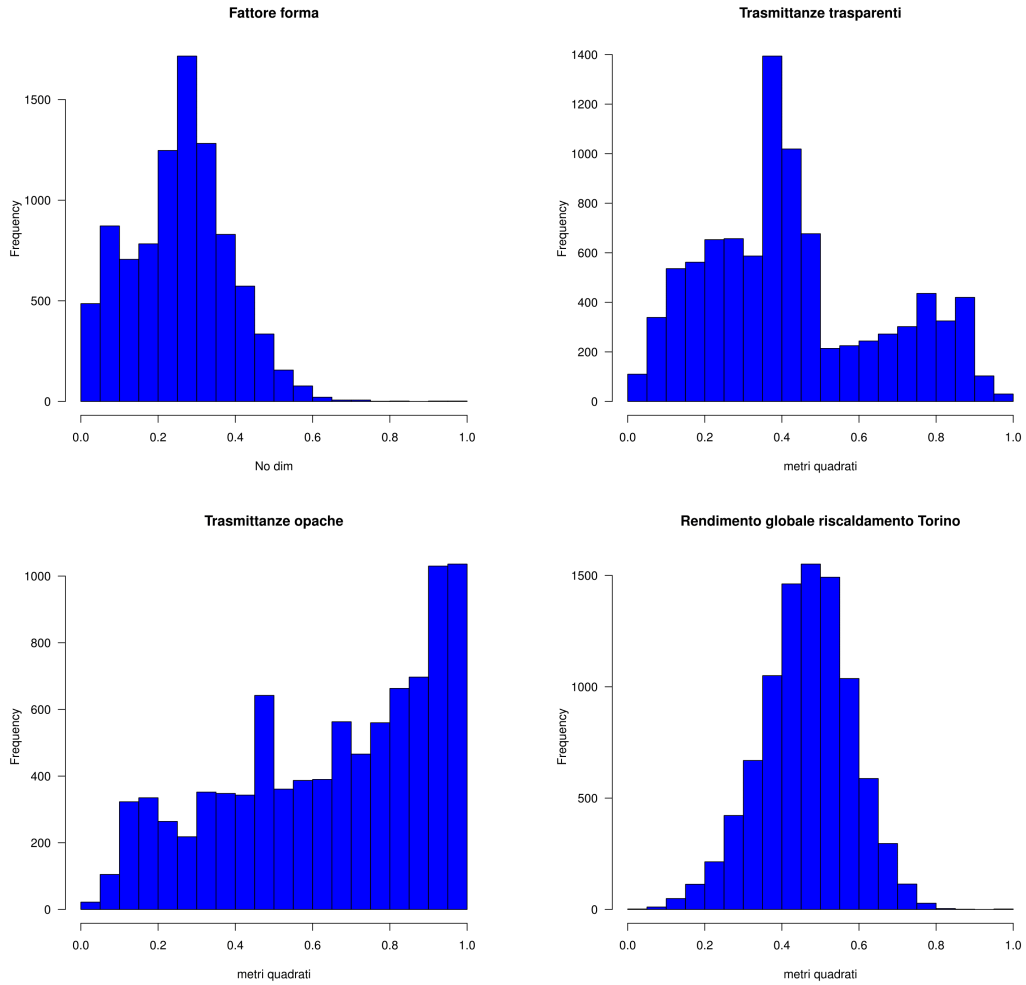


Figura 5.1: Istogrammi degli attributi principali del dataset D1

Il dataset *D*1 non contiene l'attributo *CLASSE ENERGETICA*, di conseguenza abbiamo recuperato questa informazione tramite un'operazione di *join*[2] tra il dataset *D*1 ed il dataset estratto dal catasto contenente tutte le informazioni che sono presenti nella certificazione energetica. In generale, tutte le operazioni di join tra tabelle sono state eseguite con RapidMiner.
Avendo incluso *CLASSE ENERGETICA* nel dataset *D*1, il passo successivo è stato quello di dichiarare variabile dipendente e regressori nel modello di regressione. Come abbiamo già detto nel paragrafo 4.2.1, il modello di regressione è applicabile solamente a dati numerici. L'attributo *CLASSE ENERGETICA* è invece di

[2]Il join è un'operazione che serve a combinare i record di due o più relazioni di un database tramite l'operazione di congiunzione. Utilizzando l'ID, possiamo risalire alla classe energetica di ciascun record ed includere tale attributo nel dataset D1

tipo *char*, ossia un carattere testuale. Per ovviare a questo problema, abbiamo trasformato tale attributo da testuale a numerico attribuendo un peso: la classe A+, essendo la più performante, ha il peso massimo mentre la classe G, la meno performante, ha il peso minimo. Di seguito nella tabella 5.3 possiamo notare il peso attribuito a *CLASSE ENERGETICA*:

Classe energetica	**Peso**
A+	8
A	7
B	6
C	5
D	4
E	3
F	2
G	1

Tabella 5.3: Pesi attribuiti a ciascuna classe energetica

Indichiamo con Y la variabile dipendente *CLASSE ENERGETICA* mentre con X_i con $i = 1, \ldots, 16$ ciascuno dei 16 attributi compresi nel dataset. Il modello di regressione creato è il seguente:

$Y = X_1 + X_2 + X_3 + X_4 + X_5 + X_6 + X_7 + X_8 + X_9 + X_{10} + X_{11} + X_{12} + X_{13} + X_{14} + X_{15}, + X_{16}$.

Definito il modello di regressione, possiamo procedere con l'analisi della multicollinearità (si rimanda il lettore al paragrafo 4.2.1). Utilizziamo il pacchetto *mctest* di RStudio per la sua esecuzione. Come spiegato nel capitolo 4, prima di procedere con l'analisi della multicollinearità, dobbiamo analizzare la varianza degli attributi per verificare che essa sia diversa da 0 (le implicazioni teoriche di una varianza nulla nell'analisi di regressione sono spiegate nel paragrafo 4.2.1). Di seguito, riportiamo il codice utilizzato:

```
DataColAn <- read_xlsx("C:/Users/mirko/Desktop/TESI/File_16_variabili.xlsx")
DataColAn <- [DataColAn,-17]#Esclude la response
VarAttr <- colSds(DataColAn) #calcola la varianza delle colonne
```

Creiamo un *dataframe* chiamato *DataColAn* dove includiamo tutto il dataset D1, comprendente anche l'attributo *CLASSE ENERGETICA* recuperato con l'operazione di join. Nella tabella 5.4 visualizziamo le varianze dei singoli regressori:

Come possiamo vedere nella tabella 5.4, le varianze dei regressori sono tutte positive, pertanto manteniamo ogni regressore nell'analisi.

Tabella 5.4: Varianza associata agli attributi del dataset D_1

Features	**Varianza**
X_1	0,1975
X_2	0,1668
X_3	0,1280
X_4	0,2686
X_5	0,2277
X_6	0,1986
X_7	0,2410
X_8	0,1986
X_9	0,2064
X_{10}	0,1166
X_{11}	0,1301
X_{12}	0,089
X_{13}	0,042
X_{14}	0,0087
X_{15}	0,1027
X_{16}	0,012

Il passo successivo è quello dello studio a livello *overall* della multicollinearità: come detto nel paragrafo dedicato, è utile un'analisi che verifichi la sola presenza di collinearità o meno nel modello di regressione. Il codice utilizzato è il seguente:

```
XMctest <- DataColAn #Matrice dei regressori
XMctest <- as.matrix(XMctest) #Dichiaro che è una matrice
omc <- omcdiag(XMctest,Y) #Overall multicollinearità
```

Viene creata una matrice contenente le osservazioni associate ai regressori (*XMctest*) e successivamente viene lanciato il comando *omc*, che ha bisogno come input della matrice *XMctest* e della variabile dipendente Y. Il codice precedente produce l'output riportato nella seguente tabella, la cui spiegazione nel dettaglio è riportata nel paragrafo 4.2.1:
Possiamo notare che il *Determinant*, il *Farrar-Chi-Square* ed il *Condition Number* segnalano la presenza di collinearità, pertanto approfondiamo l'analisi a livello di singolo regressore, utilizzando per prima cosa la Decomposizione di Cholesky (paragrafo 4.2.1) applicata alla matrice di correlazione dei regressori.:

```
CorrelationMatrix <- cor(DataColAn, method = "pearson")#matrice correlazione
Cholesky <- chol(CorrelationMatrix) #metodo Cholesky
```

il cui output è visualizzabile in tabella 5.6:

Tabella 5.5: Sintesi overall multicollinearity analysis

Misura	**Risultato**	**Detection**
Determinant	0,01	1
Farrar Chi-Square	4220,00	1
Red Indicator	0,19	0
Sum of Lambda Invers	3710,00	0
Theil Indicator	-8027,00	0
Condition Number	4306,00	1

	X1	**X2**	**X3**	**X4**	**X5**	**X6**	**X7**	**X8**	**X9**	**X10**	**X11**	**X12**	**X13**	**X14**	**X15**	**X16**
X1	1,00	-0,09	0,08	-0,08	-0,10	0,06	0,06	0,05	0,17	0,13	0,03	-0,24	-0,09	0,01	0,21	0,00
X2	0,00	1,00	0,06	0,01	0,01	0,14	-0,01	0,11	0,11	0,14	0,12	0,04	0,03	0,00	0,10	0,02
X3	0,00	0,00	0,99	-0,10	-0,08	0,07	0,11	-0,07	0,20	0,12	0,48	-0,09	-0,17	0,00	0,03	0,03
X4	0,00	0,00	0,00	0,99	0,44	-0,10	-0,15	-0,49	-0,15	-0,29	0,51	0,22	0,09	0,00	-0,30	-0,01
X5	0,00	0,00	0,00	0,00	0,89	-0,06	-0,11	-0,17	-0,16	-0,17	0,22	0,20	0,12	0,01	-0,13	0,00
X6	0,00	0,00	0,00	0,00	0,00	0,98	0,11	0,08	-0,10	0,66	-0,01	-0,07	0,08	-0,01	0,12	0,01
X7	0,00	0,00	0,00	0,00	0,00	0,00	0,97	0,01	0,08	0,46	0,01	-0,07	-0,07	0,01	0,02	-0,01
X8	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,84	0,09	0,20	-0,22	-0,04	0,03	0,00	0,14	-0,01
X9	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,92	0,22	-0,01	-0,15	-0,15	0,00	-0,05	-0,01
X10	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,32	0,07	-0,03	0,01	0,00	0,05	0,00
X11	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,63	-0,03	-0,04	0,02	-0,06	-0,02
X12	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,90	0,09	-0,04	-0,19	-0,05
X13	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,95	0,00	0,01	-0,07
X14	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	-0,01	0,06
X15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,87	0,02
X16	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,99

Tabella 5.6: Decomposizione di Cholesky applicata alla matrice di correlazione dei regressori del dataset D1

La funzione *cor* calcola la matrice di correlazione sfruttando le osservazioni associate ai regressori, mentre la funzione *chol* effettua la Decomposizione di Cholesky sulla matrice che viene utilizzata come input per tale funzione (nel nostro caso la matrice è *CorrelationMatrix*). Ricordiamo che utilizzando la Decomposizione di Cholesky viene rilevata collinearità se sulla diagonale principale sono presenti valori prossimi allo zero. In questo esperimento, la Decomposizione di Cholesky non ha rilevato multicollinearità, pertanto abbiamo approfondito l'analisi utilizzando la funzione *imc* del package *mctest*:

```
imc <- imcdiag(XMctest, Y)
```

Tale funzione, utilizza le misure descritte nel capitolo 4 per individuare quali regressori sono responsabili della collinearità. Riportiamo in tabella 5.7 l'output del primo run della funzione *imc* sulla matrice *XMctest* dei regressori.

	VIF	**TOL**	**Wi**	**Fi**	**Leamer**	**CVIF**	**Klein**
X1	1,130	0,885	78,026	83,609	0,941	-0,062	0
X2	1,123	0,891	73,600	78,866	0,944	-0,061	0
X3	1,767	0,566	460,404	493,345	0,752	-0,096	0
X4	1,774	0,564	464,464	497,695	0,751	-0,097	0
X5	1,523	0,657	313,875	336,332	0,810	-0,083	0
X6	4,910	0,204	2.347,025	2.514,949	0,451	-0,267	0
X7	3,023	0,331	1.214,739	1.301,651	0,575	-0,165	0
X8	2,111	0,474	667,200	714,937	0,688	-0,115	0
X9	1,788	0,559	473,300	507,163	0,748	-0,097	0
X10	9,670	0,103	5.204,751	5.577,138	0,322	-0,526	0
X11	2,523	0,396	914,090	979,491	0,630	-0,137	0
X12	1,306	0,766	183,887	197,044	0,875	-0,071	0
X13	1,122	0,892	72,978	78,200	0,944	-0,061	0
X14	1,007	0,993	3,961	4,245	0,997	-0,055	0
X15	1,320	0,758	191,922	205,654	0,870	-0,072	0
X16	1,014	0,986	8,625	9,242	0,993	-0,055	0

Tabella 5.7: Risultato della funzione *imc*

I dettagli relativi alle misure diagnostiche utilizzate dalla funzione *imc* sono presenti nel capitolo 4. I valori di soglia di VIF e TOL sono stati fissati rispettivamente a 10 e 0.1: affinché sia presente collinearità in un regressore, i valori di queste due misure devono pertanto superare rispettivamente 10 e 0.1, cosa che non accade nel nostro esempio. Pertanto, in ragione di quanto appena detto e dei valori degli altri indici, possiamo affermare che non è presente multicollinearità fra i regressori.

Questo ci permette finalmente di applicare il modello di Regressione Multipla ed Anova agli attributi del nostro dataset, per verificare quali caratteristiche sono significative e quali invece occorre tralasciare. Il codice R utilizzato per implementare il modello è il seguente:

```
reg<-lm(Y~X1+X2+X3+X4+X5+X6+X7+X8+X9+X10+X11+X12+X13+X14+X15+X16,DataCollAn)
summary(reg)
anova(reg)
```

La funzione *lm* è una funzione di base di R che consente di calcolare la regressione lineare del modello che le viene dato come input. Il primo argomento di tale funzione è il nostro modello di regressione, mentre il secondo argomento è il dataframe in cui sono presenti le osservazioni associate alla variabile dipendente ed ai regressori. La funzione *summary* riassume i risultati della regressione; la funzione *anova* ha come argomento il risultato della regressione ed applica ad esso l'analisi della varianza (per maggiori dettagli si veda il paragrafo 4.2.1). Nelle tabelle 5.8 e 5.9 mostriamo rispettivamente l'output della Regressione e dell'ANOVA.

Tabella 5.8: Output del primo run di Regressione Lineare Multipla

Termine	**Estimate**	**Std.error**	**Statistic**	**P-value**	
(Intercept)	5,057	0,329	15,371	1,2E-38	***
X1	-0,063	0,021	-2,984	2,9E-03	**
X2	-0,050	0,025	-1,992	4,6E-02	*
X3	-0,101	0,041	-2,453	1,4E-02	*
X4	-0,064	0,020	-32,489	1,2E-03	**
X5	-0,118	0,021	-5,496	4,0E+06	***
X6	0,045	0,044	10,171	3,1E-01	
X7	0,065	0,028	2,285	2,2E-02	*
X8	-0,003	0,029	-0,087	9,3E-01	
X9	0,021	0,026	0,835	4,0E-01	
X10	2,961	0,105	2,818	1,8E15	***
X11	-9,411	0,048	-1,953	0,0E+00	***
X12	-2,607	0,051	-5,150	0,0E+00	***
X13	-0,232	0,098	-2,361	1,8E-02	*
X14	0,247	0,462	0,534	5,9E-01	
X15	-0,001	0,044	-0,022	9,8E-01	
X16	0,712	0,340	2,093	3,6E-02	*
Residual std. Error	0,375				
R-squared	0,9336				
Adj R-squared	0,9335				

Nel capitolo 4, vengono illustrate nel dettaglio tutte le misure che la Regressione Lineare Multipla e l'ANOVA ci forniscono come output. In pratica, un regressore è

Tabella 5.9: Output del primo run di ANOVA sui risultati della regressione multipla

Term	Df	Sumsq	Meansq	Statistic	P-value	
X1	1	29,990	29,990	21,329	9,21E-34	***
X2	1	12,545	12,545	8,922	5,10E-173	***
X3	1	2.500,442	2.500,442	1.778,335	0	***
X4	1	5.851,774	5.851,774	4.161,829	0	***
X5	1	145,546	145,546	103,514	0	***
X6	1	643,271	643,271	457,499	0	***
X7	1	2.701,031	2.701,031	1.920,995	0	***
X8	1	1.050,100	1.050,100	746,839	0	***
X9	1	135,169	135,169	96,133	1,23E-186	***
X10	1	132,509	132,509	94,241	3,59E-08	***
X11	1	5.326,539	5.326,539	3.788,277	0	***
X12	1	3.997,053	3.997,053	2.842,736	0	***
X13	1	0,882	0,882	627,526	0,012261069	*
X14	1	0,063	0,063	0,450	0,50245643	
X15	1	0,000	0,000	0,001	0,97634523	
X16	1	0,616	0,616	438,084	0,036372942	*
Residuals	9004	127	0,140605843			

tanto più significativo nell'analisi dei dati quanto più il suo *p-value* è prossimo allo 0. In aggiunta a quanto spiegato nel paragrafo 4.2.1, l'ultima colonna delle tabelle precedenti riassume se il regressore è rilevante oppure no tramite l'uso di asterischi, indicanti i seguenti livelli di significatività:

- "***" segnala un livello di significatività pari al 100%;
- "**" segnala un livello di significatività pari al 99,99%;
- "*" segnala un livello di significatività pari al 99%;
- "." segnala un livello di significatività pari al 95%;
- " " segnala un livello di significatività inferiore al 90

È stato deciso di eliminare le variabili che presentano un livello di significatività inferiore al 90%, identificabili immediatamente poiché l'ultima colonna non presenta né asterischi né il punto. Iniziamo ad eliminare le variabili che, sia la regressione che l'ANOVA, segnalano come irrilevanti. I due metodi evidenziano che X_{14} e X_{15} sono variabili irrilevanti. Pertanto, si procede alla loro eliminazione. In primo luogo, si toglie una variabile alla volta, osservando se questa operazione comporta un innalzamento del livello della significatività dei regressori che inizialmente vengono

considerati ininfluenti. Se ciò non produce alcun effetto, allora si eliminano entrambe le variabili. A questo punto, senza i regressori X_{14} e X_{15} si esegue nuovamente la Regressione Lineare Multipla con l'obiettivo di verificare se gli attributi X_6,X_8 e X_9 abbiano acquistato rilevanza in seguito all'eliminazione di quelli precedenti. Ma anche in questo caso la regressione segnala che tali caratteristiche non sono rilevanti, quindi si è proceduto alla loro eliminazione. In definitiva, sono stati eliminati gli attributi: *Rendimento di generazione decimale*, *Rendimento di emissione decimale*, *Rendimento di distribuzione decimale*, *Rendimento medio globale stagionale acqua calda sanitaria* e *indice di prestazione energetica acqua calda sanitaria da fonti rinnovabili*. Gli attributi che rimangono dopo la Features Selection sono i seguenti:

- Superficie utile;
- Altezza media;
- Fattore forma;
- Trasmittanze opache;
- Trasmittanze trasparenti;
- Rendimento di regolazione decimale
- Rendimento globale riscaldamento Torino;
- Fabbisogno energia termica utile Torino;
- Potenza riscaldamento;
- Rendimento medio globale stagionale acqua calda sanitaria;
- Rendimento stagionale acqua calda sanitaria Torino.

Inoltre, l'elevato valore dell' R^2 e dell'$\bar{R}^2$ dimostra che il fitting del modello è molto buono. Prima di proseguire il processo di ottimizzazione con l'implementazione del DBSCAN, occorre validare i risultati ottenuti dalla Regressione Lineare Multipla e dall'ANOVA, tramite lo studio dei residui, come spiegato nel paragrafo 4.2.1. Nel nostro studio abbiamo utilizzato due strumenti grafici ed uno strumento analitico. Gli strumenti grafici utilizzati sono il plot dei residui ed il Q-Q plot.
La figura 5.2 rappresenta nella parte sinistra una nube di punti disposta casualmente nello spazio, a conferma del fatto che i residui non seguono alcun trend. Ciò verifica le ipotesi di normalità ed omoschedasticità (si rimanda il lettore al paragrafo 4.2.1). L'immagine destra presente in figura 5.2 è stata ottenuta mediante l'uso delle funzioni *qqnorm* e *qqline*. Tali funzioni plottano sullo stesso grafico i residui, disposti al crescere di n, e la bisettrice. Come già detto nel capitolo 4, più i punti si

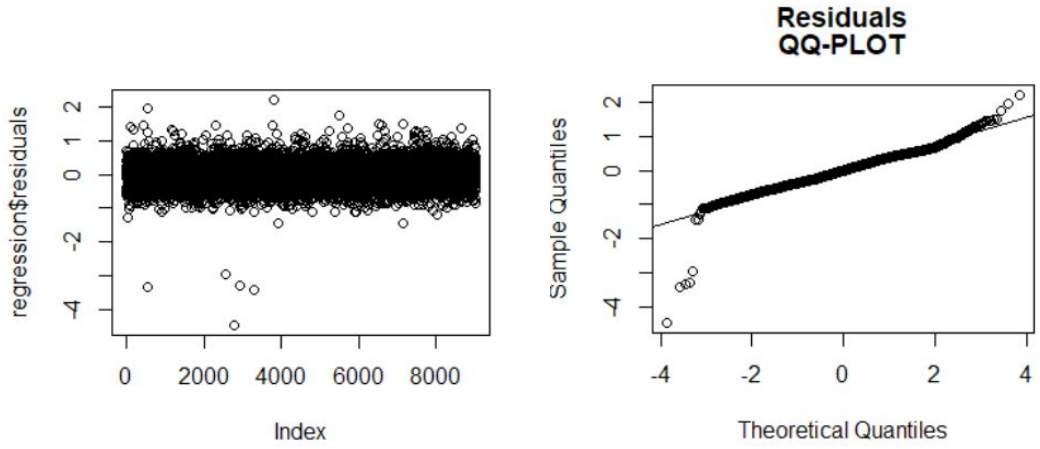


Figura 5.2: Plot e QQ-plot dei residui riferiti all'esperimento $E18$

distribuiscono lungo la bisettrice e più la distribuzione dei residui è approssimabile ad una normale. Nel nostro esperimento, la quasi totalità dei residui si dispone lungo la bisettrice: i residui sono distribuiti normalmente, di conseguenza le assunzioni sono valide. Di seguito riportiamo il codice R utilizzato per la costruzione dei grafici precedenti:

```
residuals <- regression\$residuals #Assegno i residui alla variabile residuals
#applico la funzione qqnorm ai residui e nomino il grafico
qqnorm(residuals, main = c("residuals","QQ-PLOT",
xlab = "Theoretical Quantiles", ylab = "Sample Quantiles", #nomino gli assi
plot.it = TRUE, datax = FALSE)

qqline(residuals, datax = FALSE, distribution = qnorm, #applico qqline ai residui
probs = c(0.25, 0.75), qtype = 7)
plot(residuals,xlab = "Y Estimated") #plotto il grafico qq-plot
plot(residuals) #plotto il grafico dei residui
```

Un ulteriore metodo che analizza i residui è il test di *Anderson-Darling*, che a differenza dei test precedenti è analitico. Applichiamo tale test con il seguente codice:

```
ad.test(regression$residuals)
```

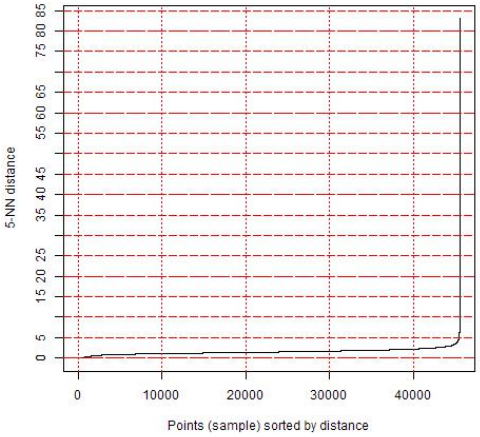
La funzione *ad.test*, calcola il p-value dei residui, nel nostro caso questo ha un valore minore di $2e^{-16}$, pertanto secondo questo test i residui sono normali.
In conclusione, data la normalità dei residui ed i valori dei parametri R^2 ed Adjusted R^2, possiamo affermare che il modello rappresenta bene i dati e che quindi è possibile togliere gli attributi menzionati prima di passare alle fasi successive.

Il blocco del framework rappresentante la Features Selection è stato eseguito solo per l'esperimento $E18$. L'esperimento $E5$ invece è stato sottoposto direttamente al blocco del DBSCAN, pertanto da questo momento in avanti, i processi che descriveremo valgono sia per $E18$ che per $E5$.
La fase successiva del *Processo di ottimizzazione dei dati* è il blocco relativo al DBSCAN. Come abbiamo già descritto nel paragrafo 4.2.2, questo algoritmo di clustering richiede due input: *eps* e *minPts*. Questi servono all'algoritmo per definire i cluster, separando i *core-point* dagli outliers. Per la scelta dei due parametri si è utilizzato il *KNN dist-plot*, unitamente ad una procedura che calcola *eps* sfruttando le derivate seconde. Infatti, per individuare l'*eps* esatto occorre capire dove il *KNN dist-plot* cambia concavità: tradotto in termini matematici, occorre calcolare in che punto la derivata seconda è positiva. Tutto questo è stato implementato in RStudio:

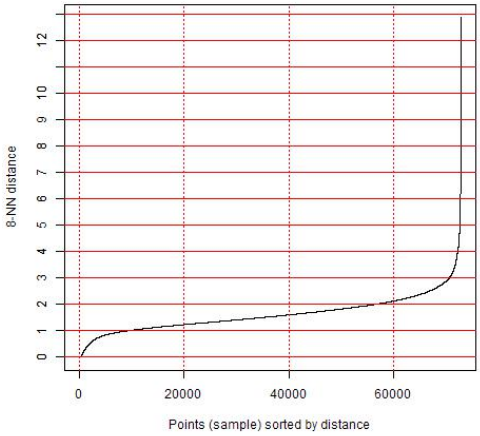
```
#Carico i dati
Data <- read_xlsx("C:/Users/mirko/Desktop/TESI/File-Dbscan.xlsx"
# Ciclo: Calcolo KNNdistance, plotto il grafico e lo salvo

for (i in c(1:20)){

kNNdistplot(DataRange ,i)
axis(2, at = seq(0,80,by = 1))
a=seq(0,80,by=1)
grid(nx = NULL, ny = NULL, col = "red", lty = "dotted")
abline(h=a,v=NULL,col="red",lty=1)
filename_k=concatenate(i,"NNdistance")
dev.copy(jpeg,filename=concatenate(filename_k,".jpeg"));
dev.off ();
KNN <- kNNdist(DataRange, i) #calcolo kNNdist
KNN <- as.data.frame(KNN) #creo un data frame per poter modellare le colonne
#ordino il data frame in ordine decrescente sull'ultima colonna
KNN <- KNN[order(KNN$`i`, decreasing=TRUE), ]
a <- KNN$`i`#assegno i valori dell'ultima colonna ad una variabile chiamata "a"
plot(a) #plotto "a" per avere nuovamente il kNNdist
axis(2, at = seq(0,80,by = 2))
axis(1, at = seq(0,10000,by = 250))
asd1=seq(0,10000,by =250)
grid(nx = NULL, ny = NULL, col = "red", lty = "dotted")
abline(h=asd1,v=NULL,col="red",lty=250)
asd2=seq(0,80,by =2)
grid(nx = NULL, ny = NULL, col = "red", lty = "dotted")
abline(h=asd2,v=NULL,col="red",lty=2)

#creo due vettori di lunghezza pari a quella di a
```

```
secondDer<-vector(mode = "numeric",length=9105)
for(i in 2:(length(a)-1)){
#assegno le derivate seconde agli elementi i del vettore "first"
secondDer[i] <- a[i+1]+a[i-1]-2*a[i]
#prendo leprime tre cifre dopo la virgola
secondDer[i] <- round(secondDer[i], digits = 3)

}#il ciclo calcola per ogni punto di "a" il valore della derivata seconda
derdiff <- diff(secondDer)
d <- order(derdiff, decreasing = TRUE)
primi50derdiff <- vector()
for(i in c(1:50)){
primi50derdiff[i] <- d[i]
}
View(primi50derdiff)
View(a)

}
```

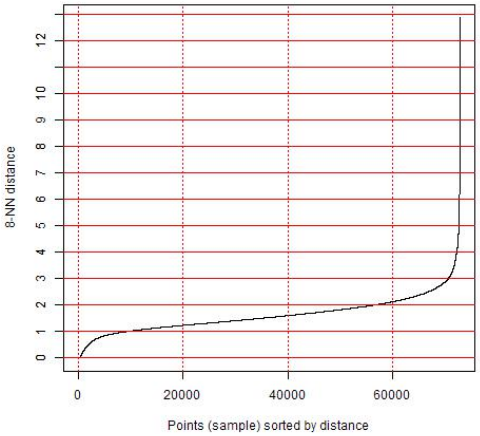
Attraverso il confronto dei vettori *primi50derdiff* e *a* e con l'analisi del *KNN dist-plot*, otteniamo che, nel primo Run di DBSCAN, per l'esperimento $E5$ i valori ottimali dei parametri sono $eps = 4.8$ e $minPts = 5$ mentre per $E18$ abbiamo ottenuto $eps = 0.32$ e $minPts = 6$. Riportiamo in figura 5.3 i *KNN dist-plots* riferiti agli esperimenti $E5$ e $E18$:



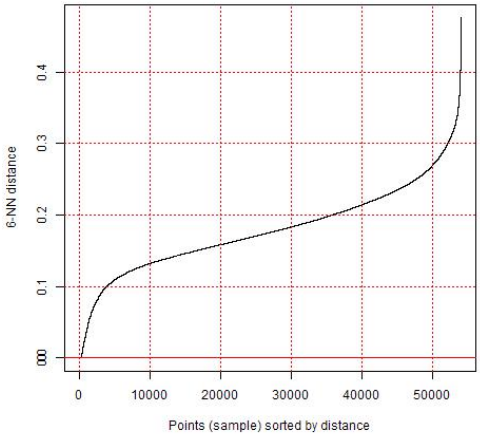
(a) *KNN dist-plot per il primo run di DBSCAN riferito all'esperimento E*5



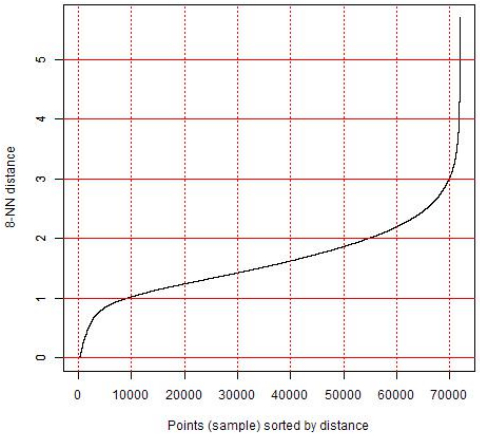
(b) *KNN dist-plot per il primo run di DBSCAN riferito all'esperimento E*18



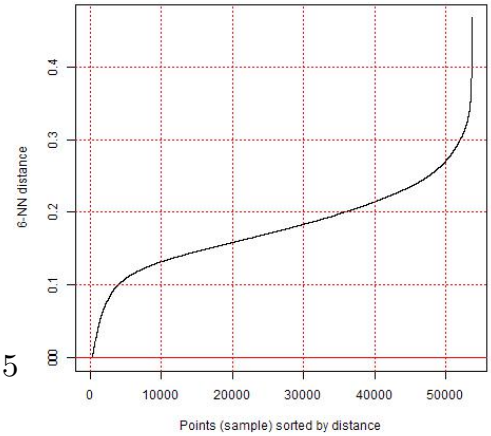
(c) *KNN dist-plot per il secondo run di DBSCAN riferito all'esperimento E*5



(d) *KNN dist-plot per il secondo run di DBSCAN riferito all'esperimento E*18



(e) *KNN dist-plot per il terzo run di DBSCAN riferito all'esperimento E*5



(f) *KNN dist-plot per il terzo run di DBSCAN riferito all'esperimento E*18

Figura 5.3: Grafici del KNN dist-plot di $E5$ ed $E18$

I valori esatti di *eps* sono stati poi calcolati con il codice R precedente che ha provveduto al calcolo delle derivate seconde. Nella tabella 5.10 vengono riassunti i valori di *eps* e *minPts* in ciascuno dei run applicati agli esperimenti: Ricordiamo

Esperimento	**Eps**	**minPts**
E5 Primo Run	4.88	5
E18 Primo Run	0.38	8
E5 Secondo Run	3.17	8
E18 Secondo Run	0.32	6
E5 Terzo Run	3.04	8
E18 Terzo Run	0.30	6

Tabella 5.10: Coppie di valori che i parametri del DBSCAN assumono in ciascuna delle tre esecuzioni, in ogni esperimento

che, come già spiegato nell'apposita sezione del capitolo 4, il valore corretto di *minPts* (K del KNN dist-plot) viene calcolato per tentativi, fin quando il grafico KNN dist-plot non subisce variazioni. Tramite il codice R precedente, abbiamo creato per ciascun run di DBSCAN 20 KNN dist-plot ed abbiamo scelto il K oltre il quale il grafico non riportava grossi cambiamenti.

Con l'ausilio del DBSCAN, abbiamo identificato ed eliminato 195 outliers nell'esperimento $E5$ e 220 nell'esperimento $E18$. I parametri del DBSCAN sono stati calcolati utilizzando le funzioni dei pacchetti disponibili in RStudio; tutte le operazioni di clustering, fra cui il DBSCAN, invece sono state eseguite su RapidMiner, che essendo un software appositamente creato per il processo di Data Mining fornisce risultati più precisi. Nella figura 5.4 mostriamo il processo utilizzato su RapidMiner per l'implementazione del DBSCAN.

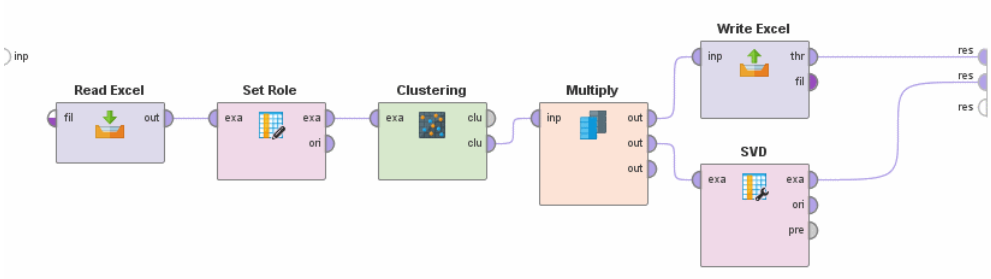


Figura 5.4: Processo RapidMiner DBSCAN

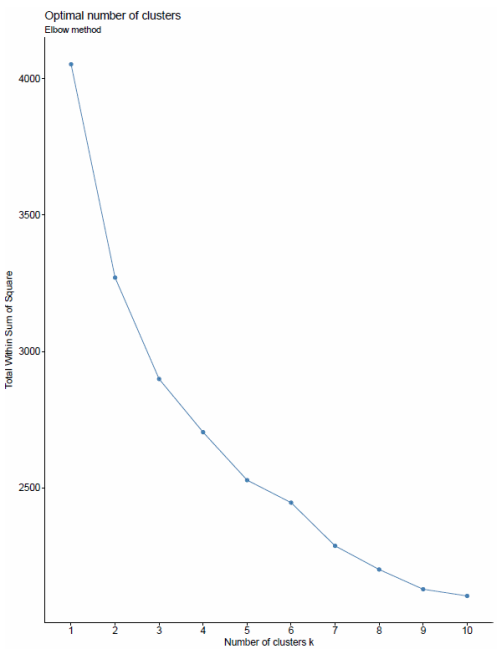
5.2 Estrazione della conoscenza: applicazione dell'algoritmo di clustering K-means

Terminato il *Processo di ottimizzazione*, i nostri esperimenti passano alla fase di *Estrazione della conoscenza* tramite applicazione del K-means, un algoritmo di clustering partizionativo. Il K-means è stato eseguito, nel caso dell'esperimento $E5$, su un dataset composto da 16 attributi, mentre nel caso di $E18$ gli attributi sono 11. Il primo passo, come è noto dalla teoria illustrata nel capitolo 4, è quello scegliere il numero di cluster con cui partizionare il dataset. Abbiamo applicato le misure menzionate nel capitolo 4, sfruttando il pacchetto *Nbclust* di RStudio. Gli argomenti dell'omonima funzione sono il dataset su cui applicare il K-means, il metodo utilizzato per il calcolo di K ed il numero di tentativi effettuati. Tale funzione è stata eseguita facendo variare K da 3 a 10. In figura 5.5 possiamo notare il grafico riassuntivo dei metodi *Elbow* e *Silhouette*; in tabella 5.11 invece vengono sintetizzati i valori ottimali di K calcolati dalle altre misure.

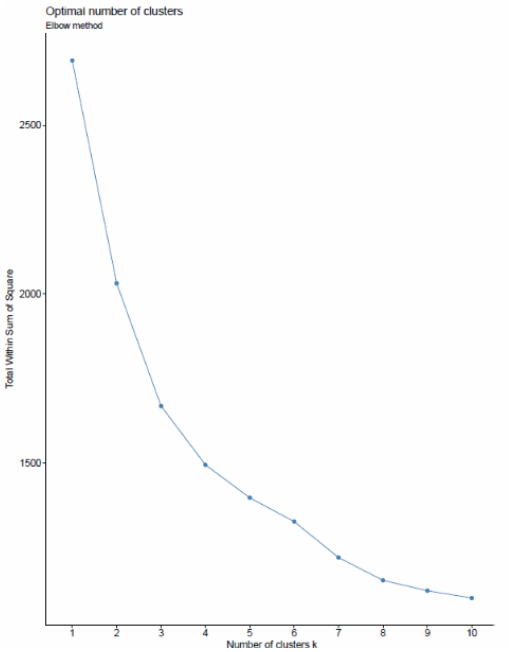
Misura	**D1,A1**	**D1,A3**
Hartigan	5	7
Scott	4	4
Trcovw	4	4
KL	6	7
Sdbw	8	8

Tabella 5.11: Riassunto dei valori di K calcolati da ogni misura utilizzata

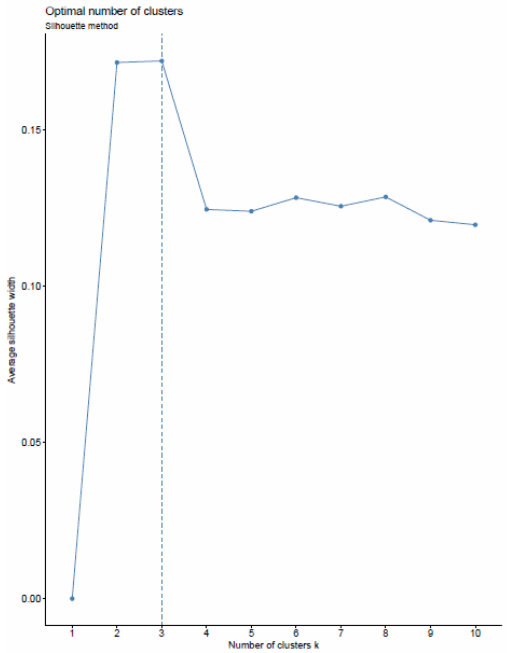
Se gli indici analitici utilizzati forniscono immediatamente il valore ottimale del parametro K, i metodi grafici utilizzano invece una tecnica euristica. Per quanto riguarda l'*Elbow*, questo analizza il grafico SSE e la scelta del parametro ricade sul numero di cluster K che massimizza la decrescenza, ovvero quel numero che rende più grande la differenza tra l'SSE calcolato tra il generico cluster i ed il cluster $i-1$: graficamente, il K suggerito è quello in cui la curva crea il gomito (dall'inglese, *Elbow*). La *Silhouette* invece è una misura sintetica della similarità fra gli oggetti appartenenti allo stesso cluster e della dissimilarità fra oggetti appartenenti a cluster differenti (si rimanda il lettore all'apposita sezione del capitolo 4, in cui viene spiegato nel dettaglio come si calcola la Silhouette): la scelta del parametro ricade sul numero di cluster K che massimizza la Silhouette.



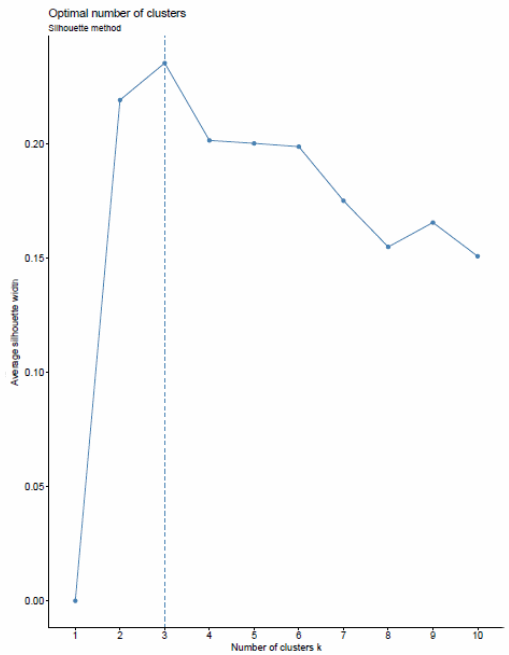
(a) *Elbow method applicato al dataset* $D1$, *con applicazione* $A1$



(b) *Elbow method applicato al dataset* $D1$, *con applicazione* $A3$



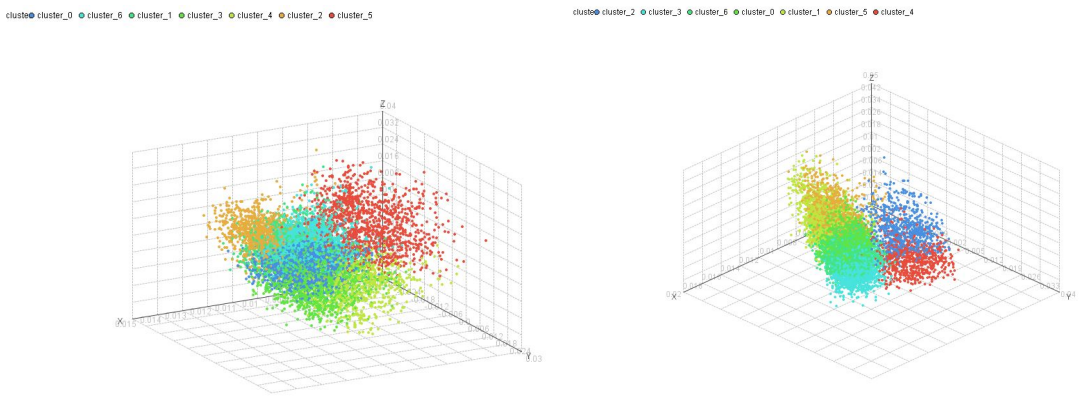
(c) *Grafico Silhouette applicata al dataset* $D1$, *con applicazione* $A1$



(d) *Grafico Silhouette applicata al dataset* $D1$, *con applicazione* $A3$

Figura 5.5: Grafici riassuntivi dell'Elbow method e della Silhouette che hanno portato alla scelta del K di $E5$ ed $E18$

Separatamente per le applicazioni $A1$ ed $A3$, i valori di K suggeriti sono stati utilizzati per partizionare il dataset $D1$. Il K-means è stato eseguito per ogni valore di K calcolato dalle misure utilizzate: ogni partizione è stata oggetto di studio mediante la caratterizzazione dei cluster, eseguita a sua volta con box-plot e tabelle che riassumono le classi energetiche contenute nei cluster stessi. Questa operazione ha consentito di individuare la performance energetica di ogni cluster. Un ulteriore metodo grafico con cui è stata valutata la partizione è il grafico a dispersione, ottenuto dopo aver applicato al dataset la *Singular Value Decomposition (SVD)*. La decomposizione a valori singolari riduce la dimensionalità a tre, in modo da poter proiettare i cluster in uno spazio tridimensionale, dove i punti appartenenti ad uno stesso cluster assumono un colore uguale. In figura 5.6 sono mostrati i grafici a dispersione relativi agli esperimenti $E5$ ed $E18$.



(a) *Grafico a dispersione dei cluster di* $E5$ (b) *Grafico a dispersione dei cluster di* $E8$

Figura 5.6: Rappresentazione mediante SVD dei cluster riferiti agli esperimenti $E5$ ed $E18$

La presente partizione è risultata la migliore fra le alternative che abbiamo calcolato. Osserviamo dal grafico a dispersione che il K-means è riuscito a partizionare bene il dataset di partenza sia nel caso in cui venga applicata la procedura $A1$, sia quando viene utilizzata la procedura $A3$. Nella figura 5.7 possiamo notare il processo progettato su RapidMiner per poter eseguire l'algoritmo K-Means.

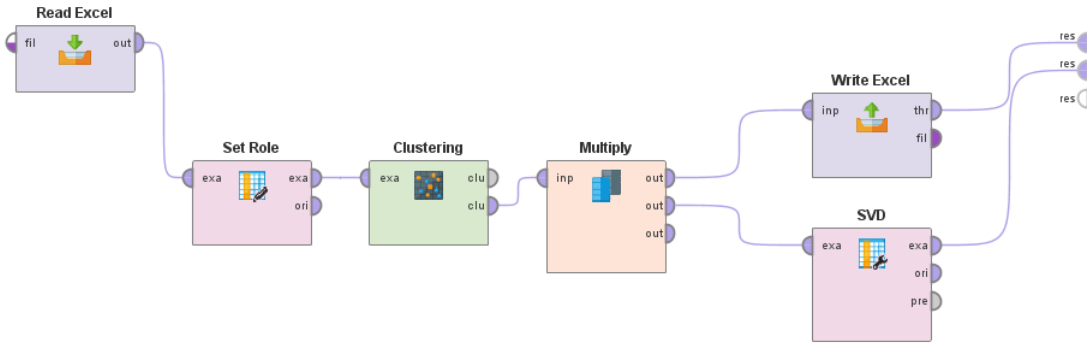


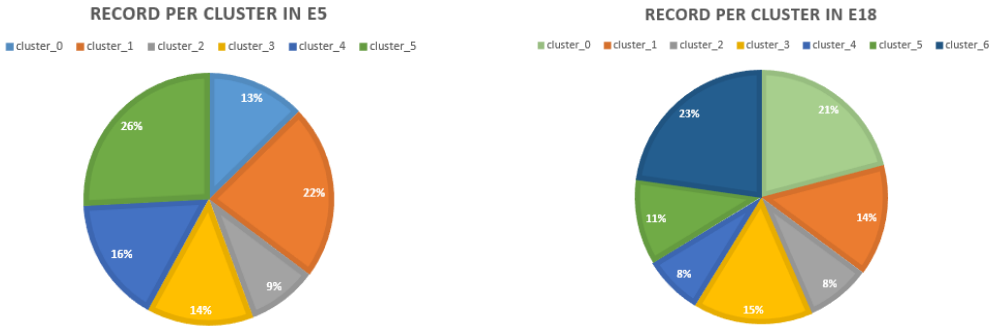
Figura 5.7: Processo RapidMiner K-means

Il risultato del K-means viene salvato in una tabella Excel differente da quella in cui è contenuto il dataset. Successivamente è stata eseguita l'operazione di *join*, tramite l'utilizzo dell'ID, per poter includere nel dataset (contenente l'etichetta di cluster) l'attributo *CLASSE ENERGETICA*, caratteristica di cui il K-means non ha tenuto conto durante l'esecuzione della partizione. Il fine ultimo del nostro esperimento è quello di dimostrare che la Classe Energetica non è una buona misura di sintesi con cui esprimere la performance energetica di un edificio: per questo non avrebbe senso includerla nel dataset da partizionare. Recuperata la caratteristica *CLASSE ENERGETICA* tramite il *join* con il dataset originale, utilizziamo in seguito i box-plot per caratterizzare i cluster e comprendere la prestazione energetica di ciascun raggruppamento. Inoltre, abbiamo creato delle tabelle riassuntive per capire se il raggruppamento degli edifici nei cluster è coerente con la classe energetica attribuita ad ogni edificio. Consideriamo un esempio: supponiamo che il cluster 1 sia quello contenente gli edifici che presentano caratteristiche termo-fisiche associate ad una performance energetica elevata, se la Classe Energetica fosse una misura di sintesi appropriata, allora il cluster 1 dovrebbe contenere edifici che possiedono prevalentemente classi energetiche A+,A e B. Su suggerimento degli esperti di dominio, abbiamo raggruppato le classi energetiche (tabella 5.12) in maniera da identificare il livello di performance energetica associato a ciascuna classe energetica:

Classi	**Performance energetica**
$\{A+, A, B\}$	Alta
$\{C, D\}$	Media
$\{E, F, G\}$	Bassa

Tabella 5.12: Raggruppamento classi energetiche e definizione della performance energetica ad esse associata

Come possiamo notare dai grafici e dalle tabelle sottostanti, la Classe Energetica non sembra essere una buona misura di sintesi per rappresentare la prestazione energetica di un edificio.



(a) *Suddivisione dei record in cluster in* $E5$ (b) *Suddivisione dei record in cluster in* $E18$

Figura 5.8: Rappresentazione dei record suddivisi per cluster

(a) *Suddivisione dei record in cluster in* $E5$(b) *Suddivisione dei record in cluster in* $E18$

Figura 5.9: Percentuale Classi Energetiche suddivise per tipologia

In figura 5.8 vediamo come sono suddivisi, in termini percentuali, i record nei vari cluster. In figura 5.9 notiamo che la percentuale di Classi Energetiche suddivise per tipologia non cambia nei due esperimenti: ciò è facilmente intuibile, dal momento che il DBSCAN ha eliminato pressoché lo stesso numero di outliers nei due esperimenti.
Nelle tabelle 5.13 vengono riportate le percentuali delle classi energetiche, suddivise per ciascun cluster. La prima tabella fa riferimento all'esperimento $E5$ mentre la seconda riporta le percentuali riferite alle classi energetiche distribuite nei cluster $E18$. Per esempio, la prima tabella ci indica che il 100% delle classi A+ si trovano tutte nel cluster 0. Invece, la tabella 5.14 individua la performance energetica degli edifici che vengono raggruppati nei cluster:
Un'ultima analisi effettuata sfrutta i box-plot per vedere le distribuzioni delle caratteristiche termo-fisiche principali contenute in ciascun cluster. Come già detto

	cluster 0	cluster 1	cluster 2	cluster 3	cluster 4	cluster 5
A+	100,00%	0,00%	0,00%	0,00%	0,00%	0,00%
A	98,08%	1,92%	0,00%	0,00%	0,00%	0,00%
B	81,76%	12,52%	2,01%	0,15%	0,46%	3,09%
C	20,93%	38,79%	11,64%	5,52%	4,97%	18,15%
D	4,15%	28,91%	10,53%	14,31%	10,61%	31,48%
E	1,27%	17,65%	7,91%	19,73%	18,22%	35,23%
F	0,17%	10,32%	8,40%	19,86%	28,70%	32,55%
G	0,11%	3,33%	8,37%	22,85%	43,67%	21,67%

	cluster 0	cluster 1	cluster 2	cluster 3	cluster 4	cluster 5	cluster 6
A+	0,00%	100,00%	0,00%	0,00%	0,00%	0,00%	0,00%
A	3,85%	96,15%	0,00%	0,00%	0,00%	0,00%	0,00%
B	12,35%	80,64%	0,76%	0,76%	0,00%	1,83%	3,66%
C	36,99%	23,42%	4,80%	6,03%	1,37%	9,55%	17,83%
D	25,85%	6,28%	10,09%	12,12%	5,88%	13,94%	25,85%
E	14,27%	2,03%	11,48%	17,95%	10,84%	13,82%	29,61%
F	10,76%	0,61%	9,80%	27,12%	13,56%	11,37%	26,77%
G	3,55%	0,11%	10,32%	36,02%	18,60%	7,74%	23,66%

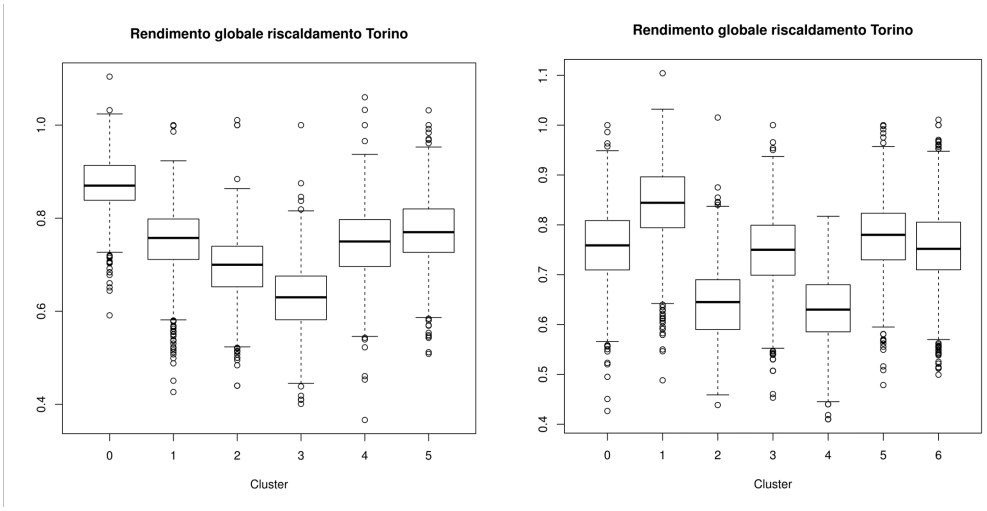
Tabella 5.13: Percentuali di classi energetiche suddivise per cluster

	cluster 0	cluster 1	cluster 2	cluster 3	cluster 4	cluster 5
$\{A+, A, B\}$	51,81%	4,10%	1,60%	0,08%	0,21%	0,87%
$\{C, D\}$	46,16%	74,51%	60,86%	37,67%	25,16%	49,48%
$\{E, F, G\}$	2,03%	21,39%	36,91%	61,19%	71,29%	49,13%

	cluster 0	cluster 1	cluster 2	cluster 3	cluster 4	cluster 5	cluster 6
$\{A+, A, B\}$	4,47%	46,59%	0,67%	0,37%	0,00%	1,24%	1,19%
$\{C, D\}$	75,01%	50,24%	46,83%	31,14%	25,74%	55,51%	49,63%
$\{E, F, G\}$	20,52%	3,17%	52,50%	68,49%	74,26%	43,25%	49,18%

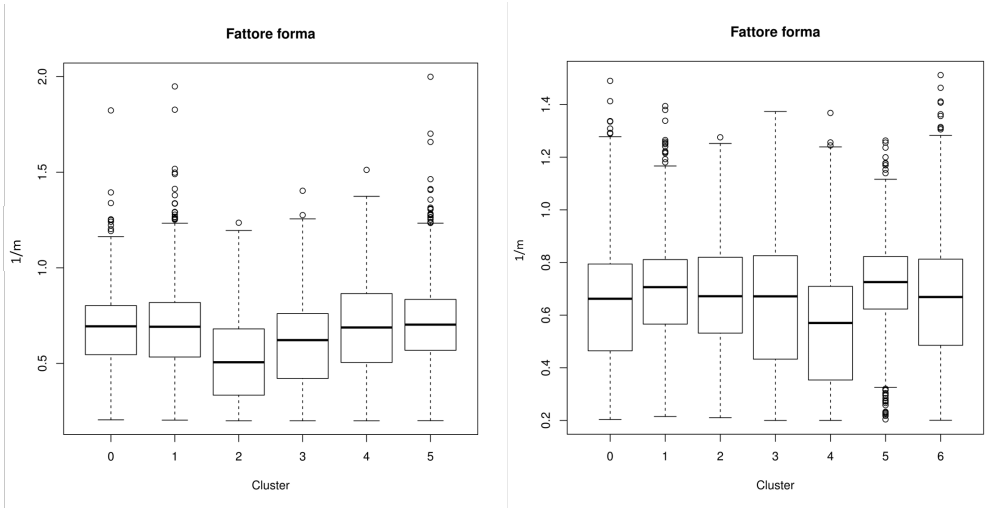
Tabella 5.14: Performance energetica edifici all'interno dei cluster

nei capitoli 1 e 2, gli esperti di dominio suggeriscono che l'efficienza energetica degli edifici viene identificata principalmente dai seguenti attributi: Fattore forma (S/V), Trasmittanze opache(U_{op}), Trasmittanze trasparenti (U_w), Rendimento globale riscaldamento Torino ($\eta_{g,To}$). I box-plot di figura 5.10 includono gli attributi appena citati raggruppati per cluster.



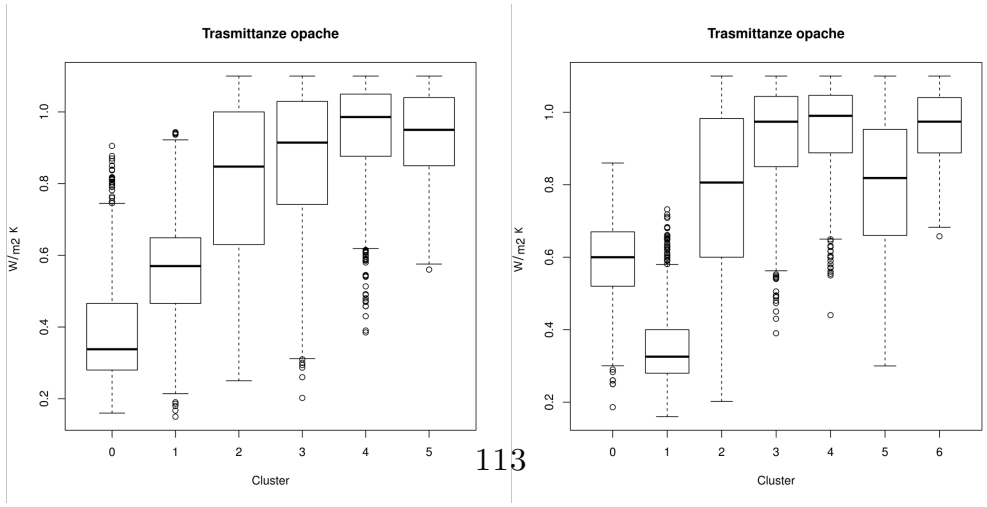
(a) *Box-plot Rendimento globale riscalda-mento To di E5*

(b) *Box-plot Rendimento globale riscalda-mento To di E18*



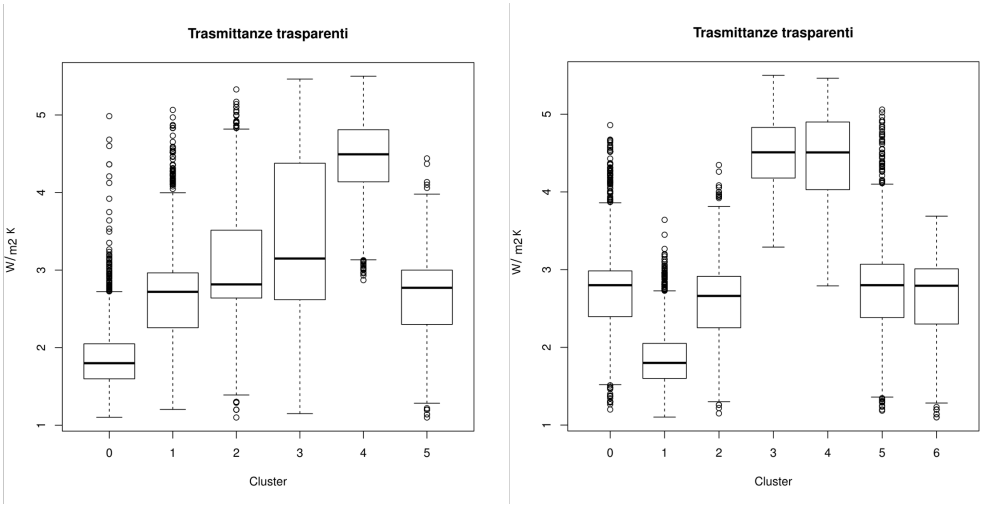
(c) *Box-plot Fattore Forma di E5*

(d) *Box-plot fattore forma di E18*



(e) *Box-plot Trasmittanze opache di E5*

(f) *Box-plot Trasmittanze opache di E18*



(g) *Box-plot Trasmittanze trasparenti di E5* (h) *Box-plot Trasmittanze trasparenti di E18*

Figura 5.10: Box-plot utilizzati per studiare la caratterizzazione dei cluster.

Le tabelle riassuntive ed i box-plot indicano la performance energetica predominante in ciascun cluster: la prima tabella si riferisce all'esperimento $E5$ mentre la seconda a $E18$. Le classi energetiche sono state raggruppate come indicato nella tabella 5.12: per esempio, la colonna *cluster 0* nella prima tabella, ci dice che in quel cluster il 51,81% degli edifici hanno una performance alta; il 46,16% hanno una performance media mentre il 2,03% hanno una performance bassa.
Il cluster 0 dell'esperimento $E5$ dovrebbe essere il cluster che raggruppa tutti gli edifici aventi un'efficienza energetica elevata: ciò in parte è confermato dalla tabella 5.13, dove notiamo che all'interno del cluster, il K-means raggruppa tutti gli edifici con classe energetica A+, quasi tutti quelli con classe energetica A e la maggior parte delle costruzioni con etichetta B. Tuttavia, anche circa il 21% degli edifici che possiedono una classe energetica C sono inclusi nel cluster 0 ed essendo più numerosi rispetto alle classi A+, A e B, questi pesano quasi queste tre ultime assieme (52% le prime tre contro il 47% delle costruzioni con classe C e D). Deduciamo che se da un lato è vero che tutti gli edifici con classi energetiche che denotano una performance alta vengono raggruppate insieme, dall'altro queste sono all'interno del cluster in cui è presente un numero elevato di costruzioni con classi energetiche rappresentanti un'efficienza media. Questo fatto si ripete anche nell'esperimento $E18$: il cluster 1 è quello ad elevata prestazione energetica, ma al suo interno è presente anche il 23% della totalità degli edifici con classe energetica C. Inoltre, osservando le tabelle 5.13 e 5.14, notiamo che casi analoghi si ripetono con altri cluster. Nei box-plot vediamo che il fattore forma è quasi uniforme in tutti i cluster. Ciò non deve destare sorpresa: il dataset di partenza include solamente edifici con destinazione d'uso E1, ossia immobili residenziali. Il fattore forma è dato dal rapporto tra la superficie

utile (S_u) ed il volume lordo (V), queste due caratteristiche geometriche negli edifici residenziali non variano molto fra loro; questo giustifica la quasi omogeneità del fattore forma in tutti i cluster degli esperimenti $E5$ ed $E18$. Negli altri box-plot, le conclusioni non sono così scontate.
Per l'esperimento $E5$, possiamo affermare che:

- il cluster 0 ha le trasmittanze opache e trasparenti più basse ed un rendimento globale di riscaldamento più alto rispetto agli altri cluster, di conseguenza concludiamo che è il raggruppamento che contiene gli edifici con una performance energetica alta. Questo è confermato dalle tabelle riassuntive, in quanto gli edifici con classi energetica A+,A,B sono tutti raggruppati in questo cluster. Notiamo però che sono presenti anche molti immobili classificati con classe energetica C;

- il cluster 1 ha le trasmittanze opache e trasparenti basse ed un rendimento globale di riscaldamento medio, paragonabile a quello dei cluster 5 e 6. Potremmo affermare che in base ai box-plot è un cluster composto prevalentemente da edifici che hanno una performance alta, in realtà consultando le tabelle riassuntive notiamo che in esso la maggioranza degli edifici presenta classi energetiche C e D;

- i cluster 2, 3 e 5 hanno trasmittanze opache alte e trasmittanze trasparenti basse. I box-plot rappresentanti tali attributi suggeriscono una mediana leggermente differente. Il rendimento globale del riscaldamento è simile per i cluster 2 e 5, ma non per il cluster 3. Quest'ultimo ha il rendimento globale del riscaldamento più basso rispetto a tutti gli altri cluster. Dai box-plot potremmo quindi concludere che i cluster 2 e 5 hanno una performance energetica media, mentre quella del cluster 3 è bassa; invece notiamo dalle tabelle riassuntive che il cluster 3 ha una maggioranza di edifici che possiedono classi energetiche C e D, mentre è il cluster 5 quello in cui viene inclusa la maggior parte di immobili con performance energetica bassa;

- il cluster 4 ha le trasmittanze opache e trasparenti più alte ed il rendimento globale del riscaldamento più basso rispetto agli altri cluster. Al suo interno, possiamo immaginare che siano inclusi prevalentemente gli edifici aventi performances energetiche basse. Questo è confermato dalle tabelle riassuntive, in cui vediamo che il 72% degli immobili presenti nel cluster 4 ha classe energetica E, F o G.

Per l'esperimento $E18$ invece deduciamo che:

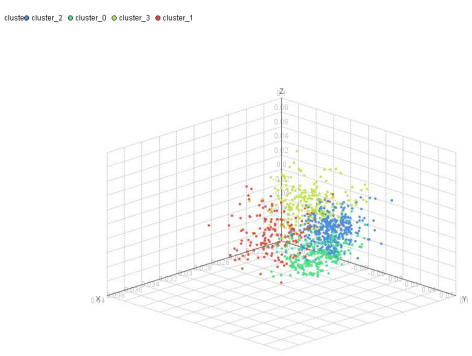
- il cluster 1 ha i valori dei tre attributi migliori rispetto agli altri cluster (trasmittanze basse e rendimento alto). Come confermato dalle tabelle riassuntive, gli immobili con una performance energetica alta sono quasi tutti raggruppati in questo cluster. Ma come già accaduto per $E5$, anche se questo è il

cluster migliore per efficienza energetica, il 50% degli edifici in esso contenuti presentano una classe energetica C o D;

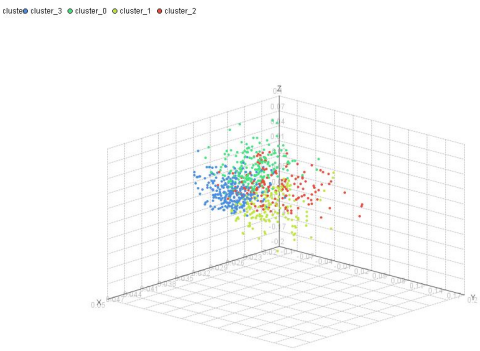
- i cluster 3 e 4 hanno trasmittanze opache e trasparenti che sono quasi identiche e sono le più alte rispetto agli altri cluster; il rendimento globale di riscaldamento è anch'esso pressoché identico ed è il più basso fra tutti i cluster. Potremmo quasi considerarli identici, il K-means ha eseguito tale partizione in quanto il fattore forma del cluster 4 è quello che si discosta maggiormente dagli altri. É evidente che dovremmo aspettarci che questi due cluster presentino al loro interno edifici con performance energetica bassa: le tabelle riassuntive lo confermano;

- i cluster 0, 2, 5 e 6 hanno un rendimento globale di riscaldamento quasi identico, più basso rispetto a quello del cluster 1 e più alto rispetto a quello dei cluster 3 e 4. Essi presentano valori molto simili anche nelle trasmittanze opache; si differenziano nelle trasmittanze trasparenti in cui il cluster 0 ha valori complessivamente più bassi rispetto a quelli dei cluster 2,5, e 6. Il cluster 6 presenta valori alti di trasmittanze opache. I box-plot suggeriscono che il cluster 0 abbia una performance energetica di livello medio-alto, i cluster 2 e 5 una performance media ed il cluster 6 una performance bassa. Le tabelle riassuntive confermano solamente i risultati del cluster 0; gli altri 3 cluster in questione invece non hanno classi energetiche ben separate.

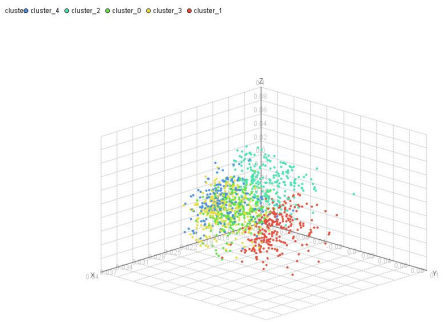
I grafici a dispersione della SVD evidenziano che in entrambi gli esperimenti il K-means ha separato bene i record. Tuttavia, i box-plot e le tabelle riassuntive portano a risultati discordanti: nelle tabelle notiamo che il K-means non è riuscito a separare bene le classi energetiche all'interno di alcuni cluster. Per esempio, i cluster ad elevata performance energetica comprendono anche una buona parte degli edifici con classe energetica C: questo dimostra che la classe energetica non è un attributo valido per poter sintetizzare l'efficienza energetica di un immobile. Tuttavia, abbiamo approfondito la nostra analisi: per essere sicuri di quanto appena affermato, gli esperimenti $E5$ ed $E18$ proseguono con lo split dei cluster in cui il K-means non è riuscito a separare le classi energetiche. Dalle tabelle riassuntive, notiamo che in $E5$ i cluster 0, 2, 3 e 5 non contengono una percentuale di edifici tale da poterli considerare cluster caratterizzati da una performance energetica precisa. Stesso discorso vale per i cluster 1, 2, 5 e 6 di $E18$. Pertanto, tramite Rapid Miner abbiamo selezionato i record appartenenti a ciascuno di questi cluster e li abbiamo inseriti in file Excel creati appositamente, uno per ogni cluster da splittare. Successivamente, è stato applicato ad ogni file Excel l'intero procedimento spiegato nel presente paragrafo. Riportiamo di seguito i risultati del grafico a dispersione relativo alla SVD, i box-plot e le tabelle riassuntive dei cluster splittati riferiti all'esperimento $E5$.



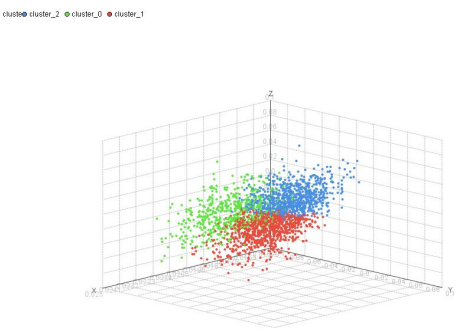
(a) *Scatter plot SVD cluster 0 di E5*



(b) *Scatter plot SVD cluster 2 di E5*



(c) *Scatter plot SVD cluster 3 di E5*



(d) *Scatter plot SVD cluster 5 di E5*

Figura 5.11: Grafici a dispersione dei cluster di E5 che sono stati splittati

Anche in questo caso, per la ricerca del valore di K ottimale, sono state utilizzate le misure di cui abbiamo parlato nel presente paragrafo. É stato poi scelto il K suggerito dal *majority model*

Split Cluster 0	**cluster 0.0**	**cluster 0.1**	**cluster 0.2**	**cluster 0.3**
$\{A+, A, B\}$	87,70%	64,21%	34,63%	23,21%
$\{C, D\}$	12,30%	34,21%	62,44%	73,21%
$\{E, F, G\}$	0,00%	1,58%	2,93%	3,57%

Split Cluster 2	**cluster 2.0**	**cluster 2.1**	**cluster 2.2**	**cluster 2.3**
$\{A+, A, B\}$	3,90%	0,00%	0,00%	1,82%
$\{C, D\}$	84,39%	52,98%	39,51%	60,73%
$\{E, F, G\}$	11,71%	47,02%	57,41%	37,45%

Split Cluster 3	**cluster 3.0**	**cluster 3.1**	**cluster 3.2**	**cluster 3.3**	**cluster 3.4**
$\{A+, A, B\}$	0,00%	0,00%	0,00%	0,00%	0,46%
$\{C, D\}$	17,51%	44,44%	23,64%	40,43%	64,22%
$\{E, F, G\}$	81,32%	53,97%	74,55%	58,87%	35,32%

Split Cluster 5	**cluster 5.0**	**cluster 5.1**	**cluster 5.2**
$\{A+, A, B\}$	0,58%	1,98%	6,17%
$\{C, D\}$	60,52%	53,84%	43,10%
$\{E, F, G\}$	38,90%	43,81%	49,85%

Tabella 5.15: Performance energetica edifici all'interno dei cluster splittati di E5

In questo caso, la generica colonna delle tabelle soprastanti è indicata come *cluster i.j*, dove i è il numero del cluster che ha subito lo split mentre j è il cluster creato dal K-means con la nuova partizione.

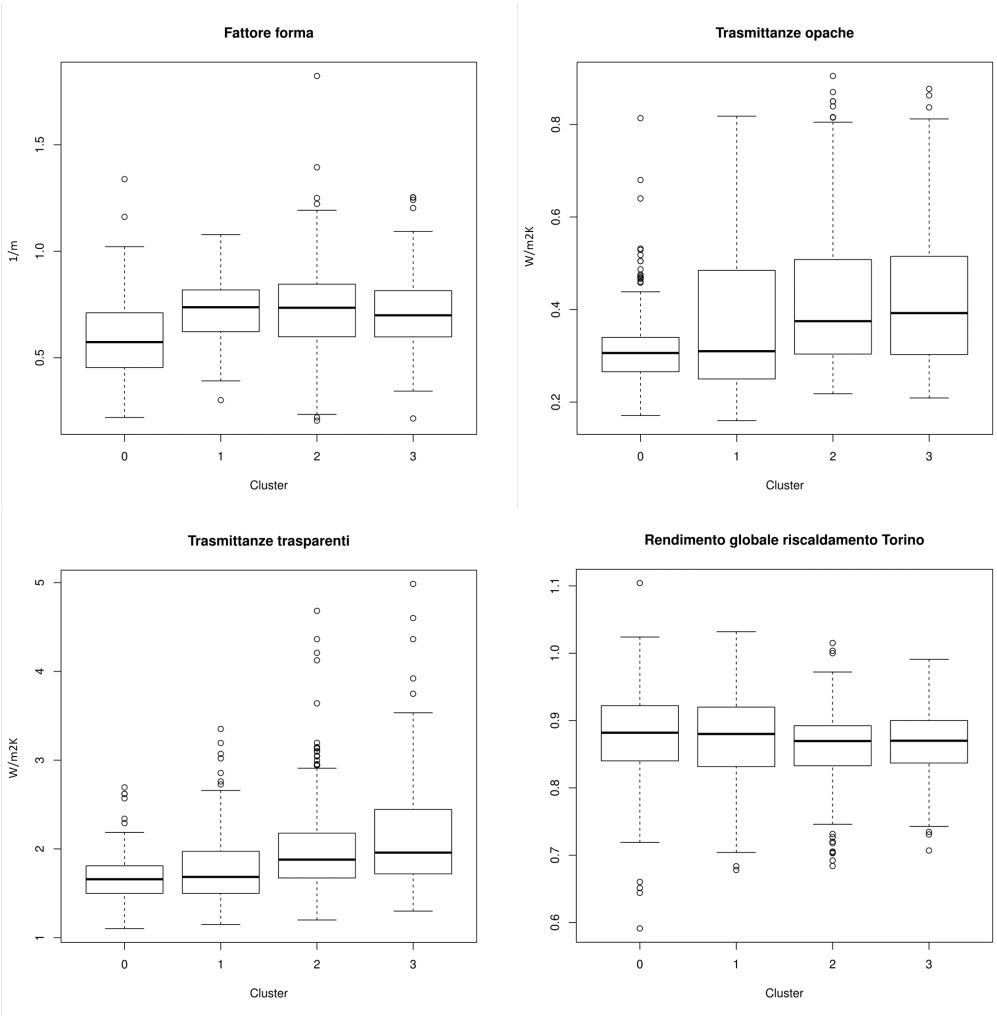


Figura 5.12: Box-plot split cluster 0 di E5

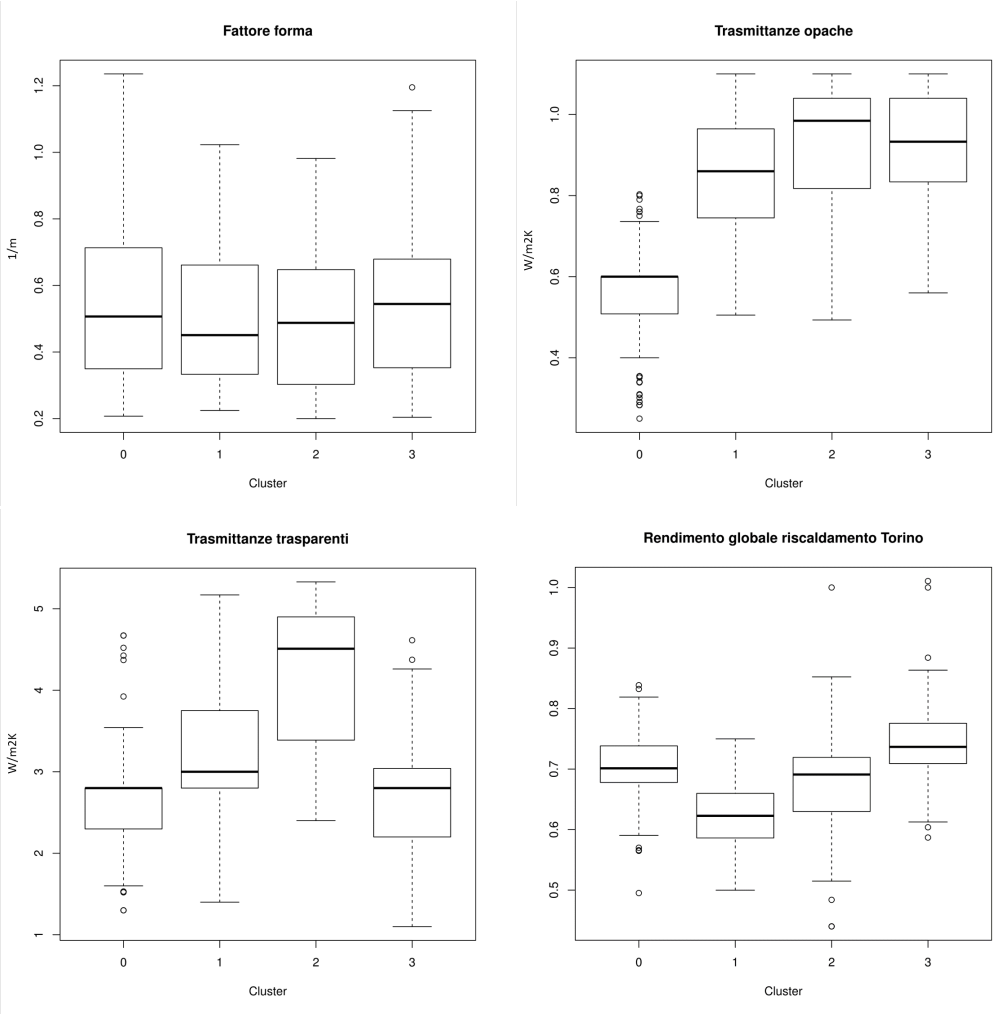


Figura 5.13: Box-plot split cluster 2 di E5

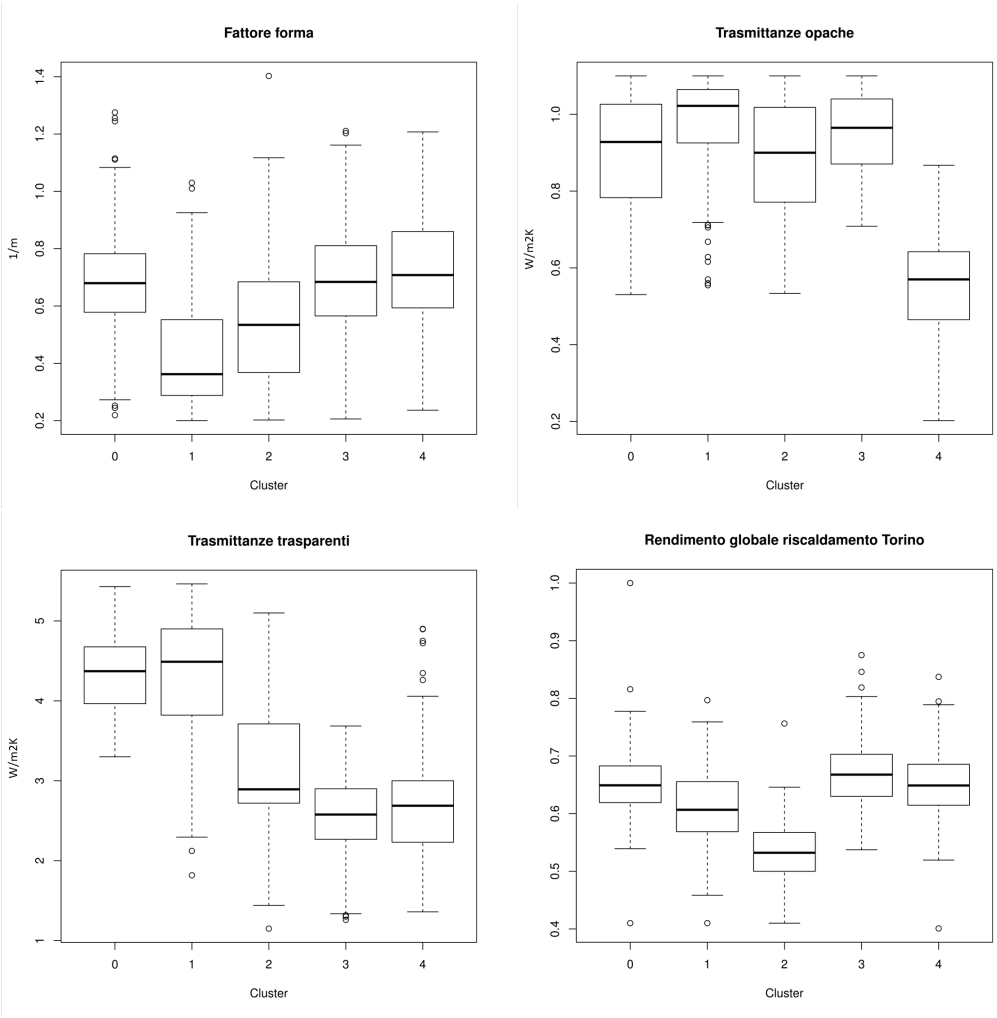


Figura 5.14: Box-plot split cluster 3 di E5

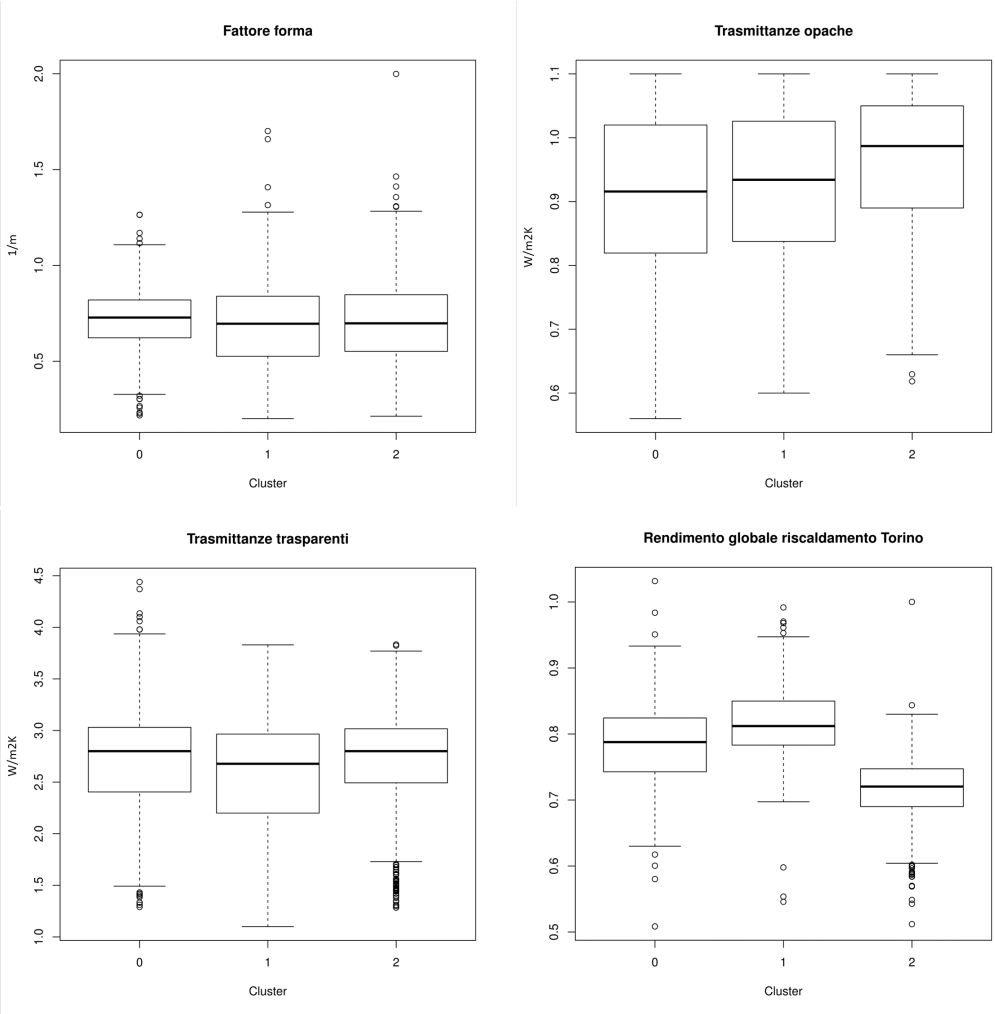


Figura 5.15: Box-plot split cluster 5 di E5

Visti i risultati dei grafici e delle tabelle precedenti, possiamo concludere che:

- il cluster 0, che ricordiamo essere quello ad alta performance energetica, non è stato partizionato meglio dalla seconda esecuzione del K-means. Dalle tabelle riassuntive deduciamo che l'algoritmo riesce a separare meglio le classi energetiche, ma tale scissione non è rilevante da un punto di vista dell'efficienza energetica. Infatti, possiamo notare dai box-plot che ad eccezione della dispersione dei valori contenuti in ciascun cluster, più o meno accentuata a seconda del gruppo in cui ogni edificio è stato ulteriormente suddiviso, la mediana dei quattro attributi è praticamente la stessa, a testimonianza del fatto che la partizione era già stata eseguita bene con il primo run di K-means;

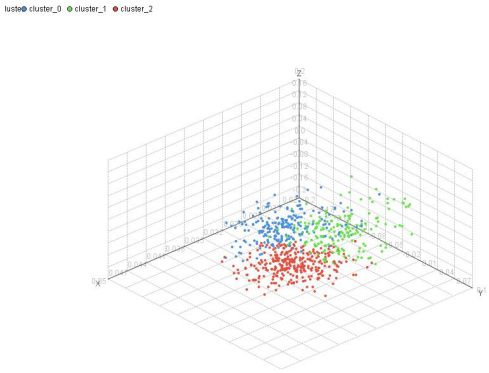
- il cluster 2, che era stato identificato come cluster a performance energetica media, è stato suddiviso in ulteriori quattro cluster. Dai box-plot notiamo

che il secondo run di K-means ha individuato che il cluster 2.0 dovrebbe essere quello con performance energetica più alta rispetto agli altri, in quanto ha entrambe le trasmittanze basse; la mediana del rendimento globale del riscaldamento è molto simile nei cluster 2.0, 2.2 e 2.3 mentre il cluster 2.1 ha rendimento energetico più basso. I box-plot quindi evidenziano che il cluster 2 ha al suo interno degli edifici aventi caratteristiche termo-fisiche discordanti da un punto di vista energetico: infatti, tipicamente un edificio che possiede trasmittanze basse ha un rendimento globale elevato; in questo caso notiamo che esistono edifici con trasmittanze basse e rendimenti bassi e con trasmittanze alte e rendimenti elevati. Questo giustifica il fatto che il K-means non sia riuscito a separare bene questi elementi nel primo run. Dalle tabelle riassuntive, notiamo tuttavia che le classi energetiche, fatta eccezione per il cluster 2.0, non sono ben separate nemmeno dopo questo ulteriore run;

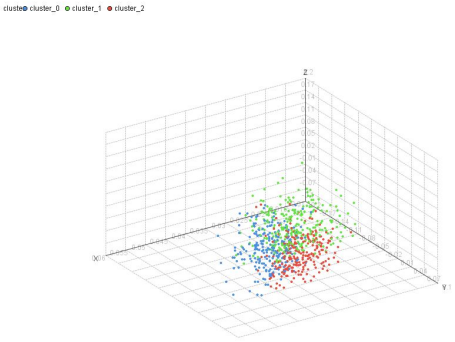
- il cluster 3, che in precedenza era stato etichettato come cluster a bassa efficienza energetica, nonostante esso includesse soprattutto edifici con classi energetiche C e D. Questo è stato suddiviso in ulteriori cinque cluster. Dai box-plot notiamo che il cluster 3.4 ha trasmittanze basse, ma un rendimento del riscaldamento paragonabile a quello degli altri cluster. In linea generale, valgono le considerazioni fatte per il cluster 2, in quanto anche in questo caso ci sono dei cluster con caratteristiche termo-fisiche discordanti da un punto di vista energetico. Inoltre, i cluster 3.0 e 3.2 sono quelli in cui vengono raggruppati gli edifici con classi energetiche E, F, G; ciononostante, alcuni attributi sono paragonabili con quelli misurati per gli immobili contenuti negli altri cluster;

- il cluster 5 era stato etichettato come cluster a performance energetica media. É stato suddiviso in ulteriori 3 cluster, ma possiamo notare dai box-plot che le caratteristiche termo-fisiche da essi possedute sono praticamente identiche, ad eccezione del rendimento globale del riscaldamento. Probabilmente, il K-means ha partizionato tenendo conto soprattutto di quest'ultimo attributo. Tra i cluster splittati, questo è quello per il quale l'algoritmo ha fatto fatica a creare nuovi raggruppamenti significativi, sia per quanto riguarda le classi energetiche che per le caratteristiche dei palazzi.

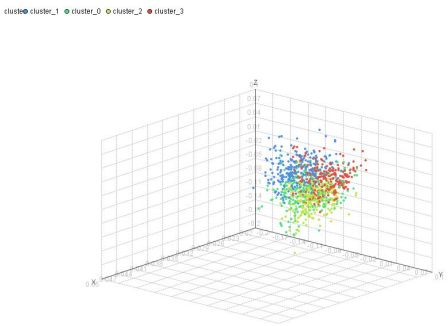
Riportiamo ora le stesse tabelle per gli split dei cluster che sono stati eseguiti nell'esperimento $E18$.



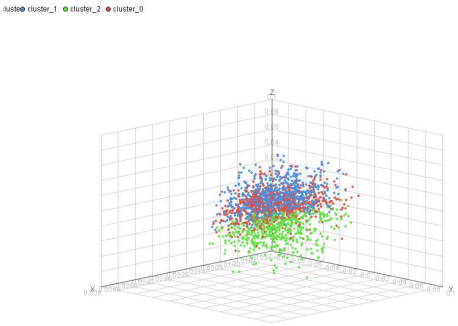
(a) *Scatter plot SVD cluster 1 di E18*



(b) *Scatter plot SVD cluster 2 di E18*



(c) *Scatter plot SVD cluster 5 di E18*



(d) *Scatter plot SVD cluster 6 di E18*

Figura 5.16: Grafici a dispersione dei cluster di E18 che sono stati splittati

Split Cluster 1	**cluster 1.0**	**cluster 1.1**	**cluster 1.2**
$\{A+, A, B\}$	52,17%	44,29%	43,92%
$\{C, D\}$	45,01%	52,92%	52,35%
$\{E, F, G\}$	2,81%	2,79%	3,73%

Split Cluster 2	**cluster 2.0**	**cluster 2.1**	**cluster 2.2**
$\{A+, A, B\}$	0,86%	1,13%	0,00%
$\{C, D\}$	18,45%	61,28%	58,26%
$\{E, F, G\}$	80,69%	37,59%	41,74%

Split Cluster 3	**cluster 3.0**	**cluster 3.1**	**cluster 3.2**	**cluster 3.3**
$\{A+, A, B\}$	0,46%	0,00%	0,00%	1,11%
$\{C, D\}$	32,87%	79,10%	45,06%	62,96%
$\{E, F, G\}$	66,67%	20,90%	54,94%	35,93%

Split Cluster 6	**cluster 6.0**	**cluster 6.1**	**cluster 6.2**
$\{A+, A, B\}$	0,00%	1,19%	2,49%
$\{C, D\}$	8,25%	67,22%	68,86%
$\{E, F, G\}$	91,75%	31,59%	28,65%

Tabella 5.16: Performance energetica edifici all'interno dei cluster splittati di E18

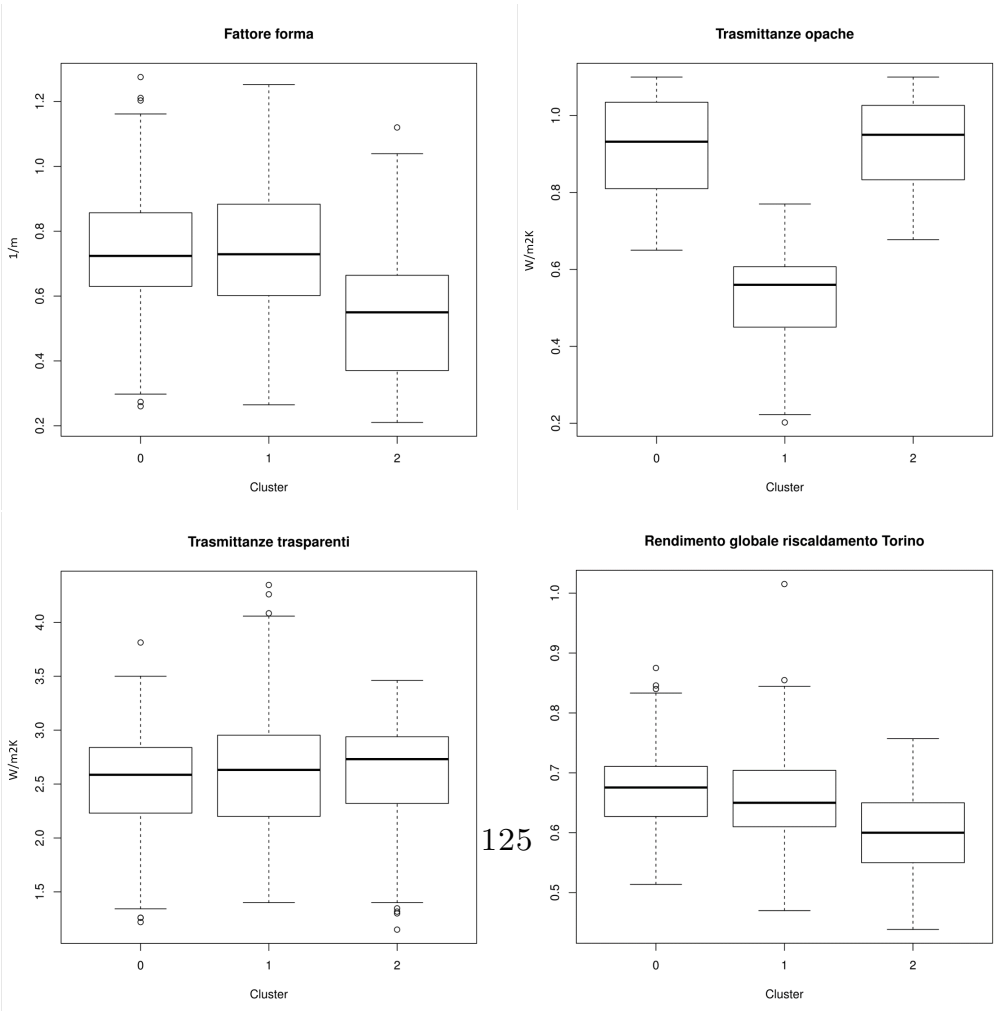


Figura 5.18: Box-plot split cluster 2 di E18

[H]

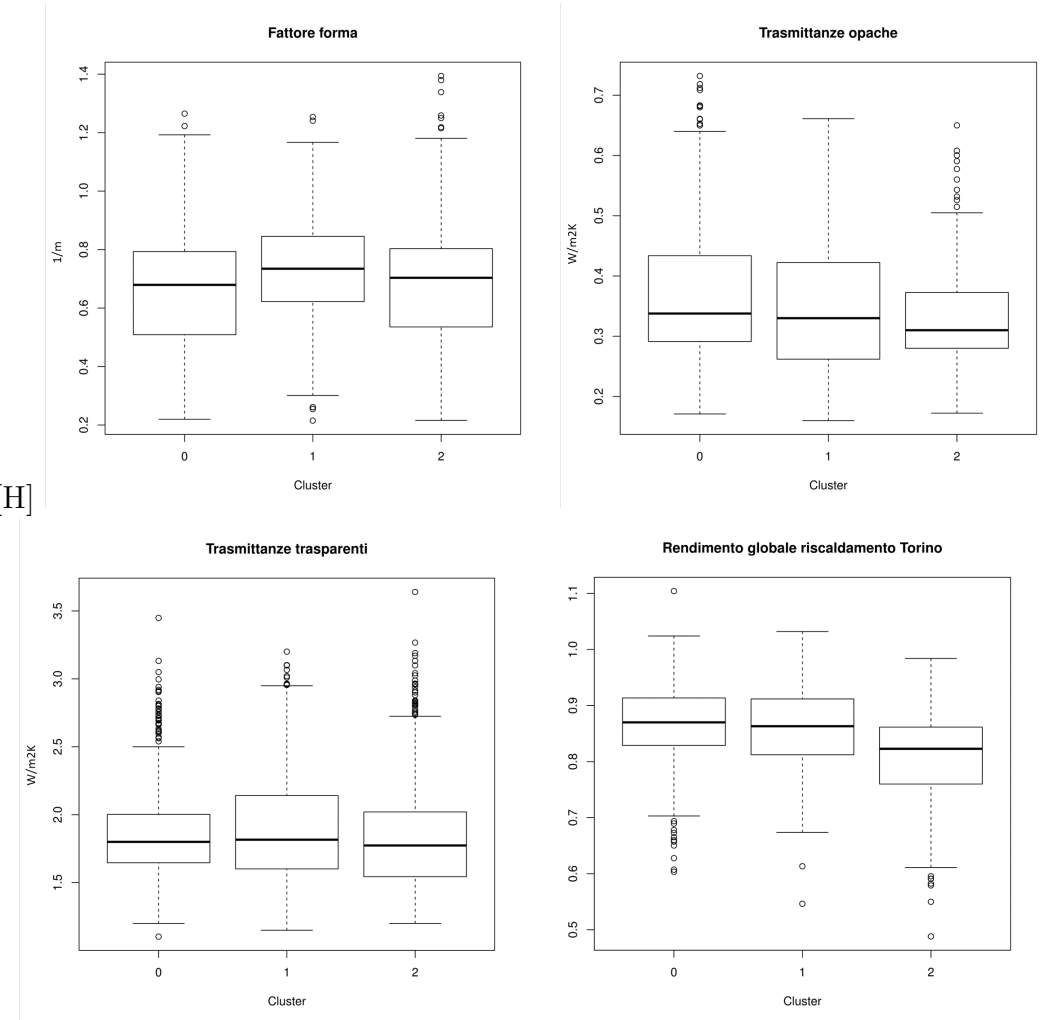
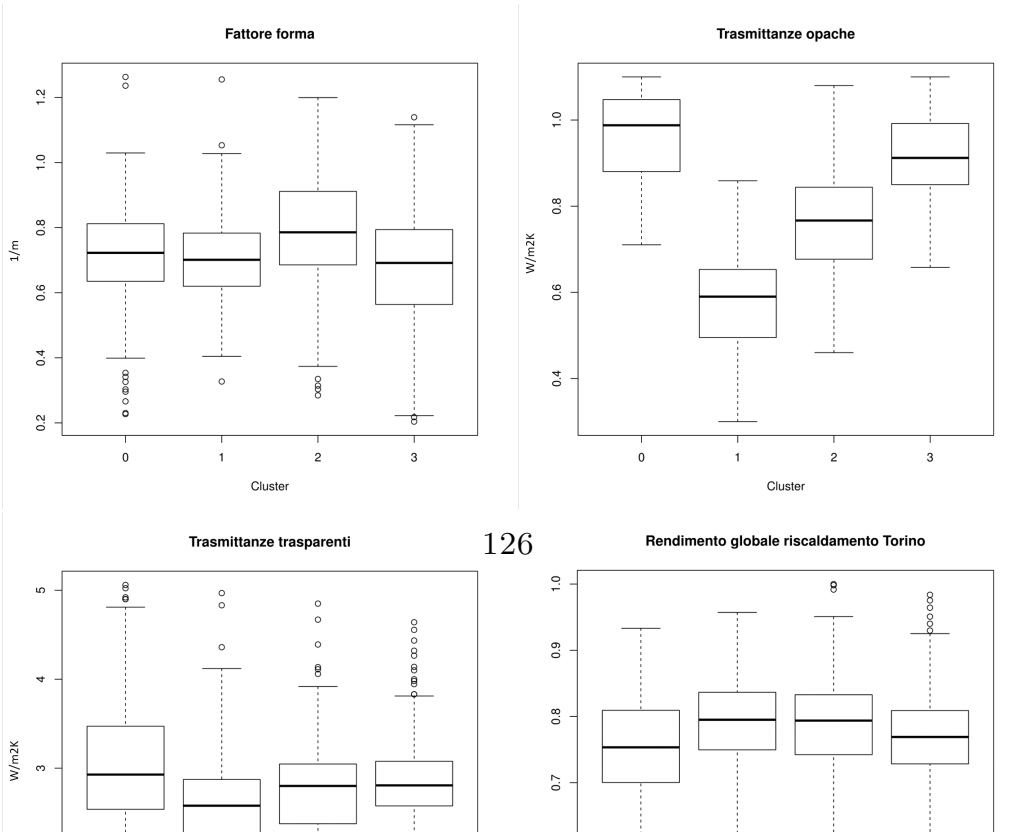


Figura 5.17: Box-plot split cluster 1 di E18



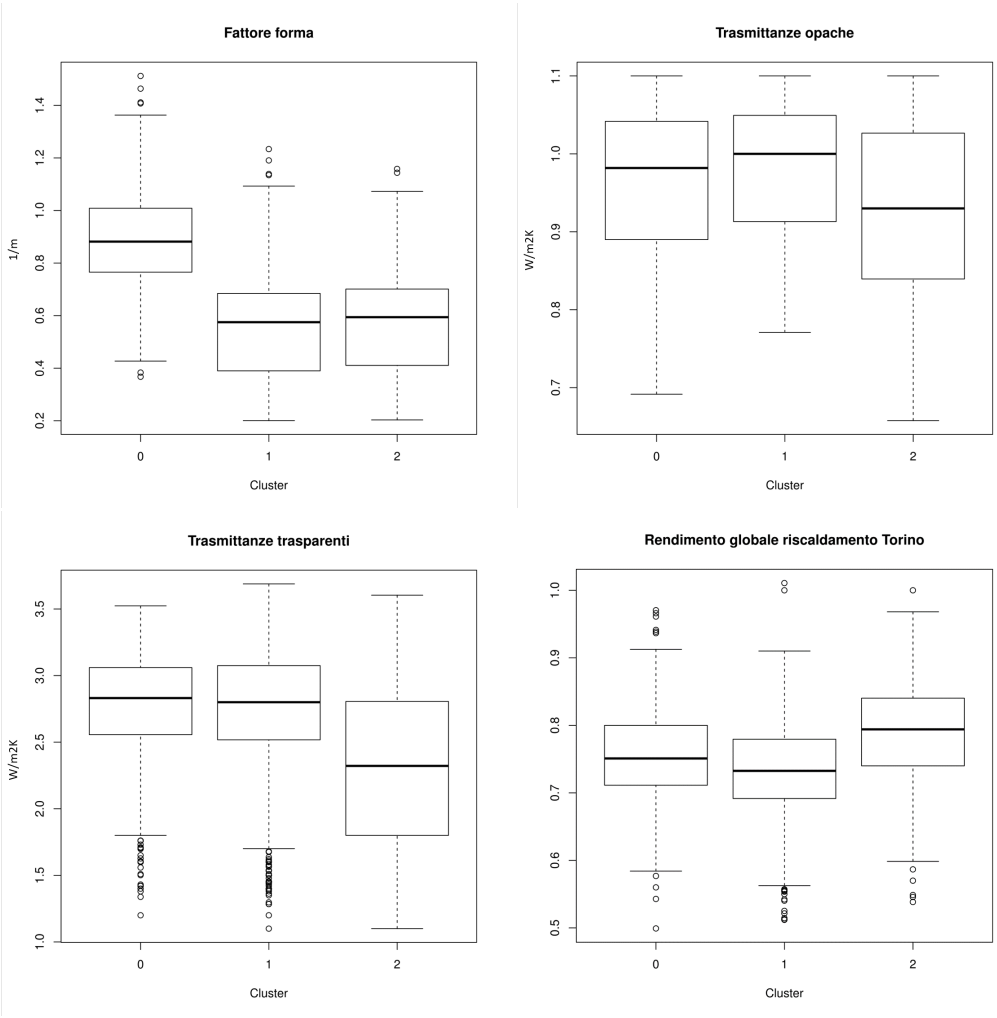


Figura 5.20: Box-plot split cluster 6 di E18

Dati i risultati dei grafici e delle tabelle soprastanti riferiti all'esperimento $E18$, possiamo dedurre che:

- il cluster 1, che abbiamo etichettato come cluster contenente gli immobili efficienti da un punto di vista energetico, è stato splittato in 3 differenti cluster, ciascuno dei quali nuovamente poco separato: le classi A+,A e B sono presenti in tutti i nuovi cluster, avremmo preferito che le classi energetiche A+ ed A fossero tutte all'interno di un unico cluster (o quantomeno una situazione simile). Anche i box-plot evidenziano un'evidente similarità: le mediane sono tutte posizionate alla stessa altezza;

- il cluster 2, identificato come raggruppamento con immobili aventi performances energetiche di livello medio, è stato partizionato in tre nuovi cluster. Il cluster 2.0 è l'unico che è stato separato bene da un punto di vista delle

classi energetiche, come vediamo in tabella riassuntiva: esso include edifici a bassa efficienza energetica. Tuttavia, i box-plot evidenziano come anche in questo caso le caratteristiche termo-fisiche siano praticamente identiche nei tre cluster, fatta eccezione per il cluster 2.1 che presenta un valore medio delle trasmittanze opache più basso rispetto agli altri;

- il cluster 5, identificato come partizione avente al suo interno costruzioni con efficienza energetica media, è stato splittato in quattro cluster differenti. I cluster 3.0 e 3.1 hanno rispettivamente una dominanza di classi $\{E, FeG\}$ e $\{C, D\}$, mentre i restanti due sono poco separati. Ma con i box-plot vale nuovamente quanto detto sopra;

- il cluster 6, infine, associato ad una performance bassa, è stato partizionato in tre ulteriori cluster. I box-plot evidenziano che questi sono distinti in base alle caratteristiche termo-fisiche, anche la rispettiva tabella riassuntiva mostra una buona separazione. Tuttavia, i risultati sono discordanti: il cluster 6.0 dovrebbe avere i valori medi delle caratteristiche termo-fisiche nettamente inferiori a quelli degli altri due cluster ma, ad eccezione del fattore forma, le trasmittanze ed il rendimento sono paragonabili

Dalle evidenze empiriche arriviamo a due conclusioni principali:

1. L'intero procedimento che partiziona i dataset esegue il proprio lavoro in maniera corretta, come testimoniano i grafici a dispersione dell'SVD ed i box-plot delle caratteristiche termo-fisiche;

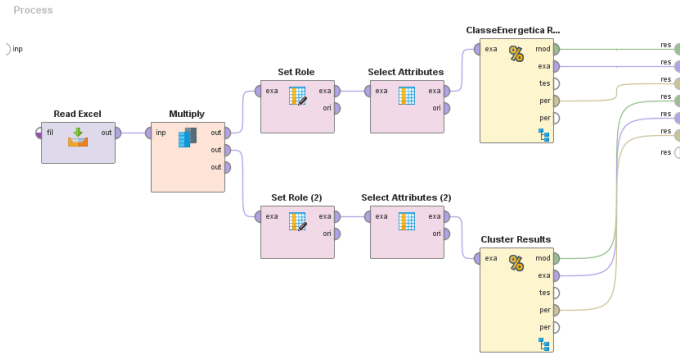
2. La classe energetica potrebbe non essere considerata un attributo che sintetizza in maniera corretta la performance energetica di un edificio.

Per dimostrare tali conclusioni, utilizziamo una tecnica di *cross-validation*: il classificatore ad albero.

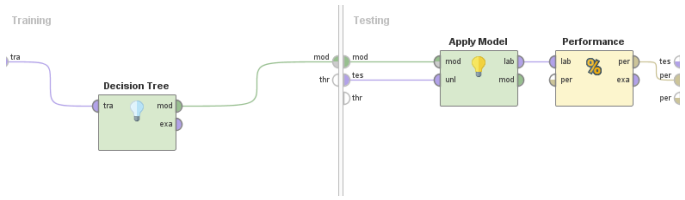
5.3 Estrazione della conoscenza: cross-validation con il classificatore ad albero.

I metodi di classificazione consentono di assegnare gli oggetti ad una classe predefinita dopo che essi ne analizzano le caratteristiche. L'obiettivo è quello di verificare la bontà del modello di clustering utilizzato ed al tempo stesso valutare se i cluster che abbiamo ottenuto dalle partizioni possono essere considerati delle migliori *label* rispetto all'attributo *CLASSE ENERGETICA*. In altre parole, attraverso il classificatore ad albero dimostreremo la veridicità delle conclusioni enunciate alla fine del paragrafo precedente. Inoltre, utilizzando le misure di performance del

classificatore, verificheremo che l'applicazione della *Features Selection* prima dell'esecuzione del *DBSCAN*, conduce a risultati migliori rispetto al solo utilizzo del *DBSCAN*. Faremo riferimento ai concetti teorici trattati nei paragrafi 2.3.2 e 4.3.2. L'algoritmo utilizzato dal nostro albero di classificazione è il C4.5, poiché abbiamo a che fare con attributi continui (l'algoritmo ID3 si utilizza solamente nel caso di attributi discreti). Nella figura sottostante, vediamo il processo di Rapid Miner con cui abbiamo costruito il classificatore ad albero.



(a) *Processo principale*



(b) *Sotto-processo*

Figura 5.21: Processo di RapidMiner con cui viene creato il classificatore ad albero

L'operatore *Multiply*, consente di dividere il processo a partire da uno stesso blocco. In questo modo, creiamo un classificatore utilizzando come *label* la classe energetica da un lato ed i cluster dall'altro. Le misure di performance utilizzate sono la precisione, l'accuratezza ed il recall: maggiore è il valore di tali indici e migliore è il modello utilizzato (si rimanda il lettore al paragrafo 4.3.2, dove vi è la spiegazione relativa al calcolo di tali misure di performance). Gli indici vengono calcolati utilizzando le matrici di confusione, ovvero delle matrici quadrate che hanno per righe e per colonne rispettivamente le classi previste e le classi effettive. Ricordiamo che nella diagonale sono presenti gli items interpretati in maniera corretta dai modelli; gli altri sono errori che i modelli commettono nel prevedere la corretta collocazione degli items nelle varie classi. Riportiamo per l'esperimento $E5$, le matrici di confusione riferite al modello che utilizza la classe energetica come label ed al modello che utilizza il cluster come label. Dalla matrice di confusione riferita al cluster dell'esperimento $E5$, ricaviamo le seguenti misure:

	true C0	**trueC6**	**trueC1**	**trueC3**	**trueC4**	**trueC2**	**trueC5**	**precision**
pred.C0	1630	109	67	116	63	26	23	80.14%
pred.C6	84	1460	91	19	21	131	3	80.71%
pred.C1	44	10	648	11	0	18	11	87.33%
pred.C3	15	21	35	1171	57	2	21	88.58%
pred.C4	17	18	11	16	589	33	7	85.24%
pred.C2	27	115	54	1	3	852	4	80.68%
pred.C5	16	17	31	26	20	14	1132	90.13%
recall	88.93%	83.43%	69.16%	86.10%	78.22%	79.18%	94.25%	

Tabella 5.17: Matrice di confusione di $E5$ riferita al cluster

- $accuracy = 83.53\%$
- $precision = 84.07\%$
- $recall = 82.75\%$

	true C	**true E**	**true D**	**true F**	**true B**	**true G**	**true A+**	**true A**	**precisi**
pred. C	1657	52	642	3	447	2	0	12	58.86%
pred. E	0	775	177	602	0	190	0	0	44.44%
pred. D	333	600	1637	140	6	9	0	0	60.07%
pred. F	0	153	3	322	0	126	0	0	53.22%
pred. B	20	0	0	0	189	0	0	11	85.91%
pred. G	0	1	0	76	0	600	0	0	82.64%
pred. A+	0	0	0	0	0	0	7	0	100.00%
pred. A	1	0	0	0	5	0	4	29	74.36%
recall	82.40%	49.02%	66.57%	28.17%	29.21%	64.38%	42.86%	55.77%	

Tabella 5.18: Matrice di confusione di E5 riferita alla classe energetica

Dalla matrice di confusione riferita alla classe energetica dell'esperimento $E5$, ricaviamo le seguenti misure:

- $accuracy = 58.81\%$
- $precision = 62.09\%$
- $recall = 49.13\%$

In seguito, mostriamo i risultati della cross-validation relativa all'esperimento $E18$.

Dalla matrice di confusione riferita al cluster dell'esperimento $E18$, ricaviamo le seguenti misure:

	trueC2	**trueC3**	**trueC6**	**trueC0**	**trueC1**	**trueC5**	**trueC4**	**precision**
pred.C2	713	0	5	5	1	4	41	92.72%
pred.C3	0	1344	9	36	0	15	2	94.92%
pred.C6	8	14	1894	81	1	17	4	93.81%
pred.C0	1	10	76	1671	283	66	0	79.31%
pred.C1	1	0	0	52	949	17	0	93.13%
pred.C5	3	8	51	12	26	849	3	89.18%
pred.C4	20	9	0	0	0	0	630	95.02%
recall	95.58%	97.04%	93.07%	89.98%	75.32%	87.35%	91.30%	

Tabella 5.19: Matrice di confusione di $E18$ riferita al cluster

- $accuracy = 90.02\%$
- $precision = 91.23\%$
- $recall = 90.05\%$

	trueE	**trueC**	**trueD**	**trueF**	**trueB**	**trueG**	**trueA+**	**trueA**	**prec**
pred.E	832	0	157	600	0	169	0	0	47.33
pred.C	53	1799	736	5	327	2	0	6	61.44
pred.D	555	197	1568	102	0	3	0	0	64.66
pred.F	134	0	6	348	0	125	0	0	56.77
pred.B	1	45	1	0	324	0	0	14	84.16
pred.G	2	0	0	88	47	627	0	0	82.07
pred.A+	0	0	0	0	0	0	4	1	80.00
pred.A	0	0	0	0	5	0	4	31	77.50
recall	52.76%	88.14%	63.53%	30.45%	49.39%	67.42%	32.86%	59.62%	

Tabella 5.20: Matrice di confusione di $E18$ riferita alla classe energetica

Dalla matrice di confusione riferita alla classe energetica dell'esperimento $E18$, ricaviamo le seguenti misure:

- $accuracy = 62.11\%$
- $precision = 67.06\%$
- $recall = 53.78\%$

É evidente che le misure utilizzate per valutare il modello sono migliori nel caso in cui si utilizzi il cluster come label piuttosto che la classe energetica. Dalle matrici di confusione capiamo perché i risultati del K-means sono contrastanti quando confrontiamo i box-plot con le tabelle contenenti le percentuali relative alle classi

energetiche contenute in ogni cluster. La matrice di confusione riferita alla classe energetica, sia per l'esperimento $E5$ che $E18$, mostra che una buona parte degli immobili aventi classe energetica C (circa il 27%) vengono predetti come costruzioni di classe energetica B. Da ciò, capiamo perché all'interno del cluster ad elevata performance erano presenti edifici con classe energetica C: non è il K-means a partizionare male, ma sono le classi energetiche che non esprimono correttamente la performance energetica desumibile dalle caratteristiche termo-fisiche di ciascun immobile. Tali considerazioni sono valide in entrambi gli esperimenti, per differenti classi energetiche: per esempio, nella tabella 5.20 vediamo che 555 classi energetiche D sono predette come classi energetiche E.
I risultati del classificatore evidenziano che l'output del clustering è più corretto rispetto a quello delle classi energetiche. Inoltre, possiamo notare come le misure associate alla matrice di confusione in tabella 5.19 siano di diversi punti percentuali migliori rispetto a quelle associate alla matrice di confusione di tabella 5.17. I risultati sottolineano una precisione addirittura oltre il 91%, risultato sorprendente che attesta la robustezza del processo di ottimizzazione costituito dal binomio Features-Selection/DBSCAN.
Il classificatore ha anche un'altra utilità. Infatti, oltre alla costruzione delle matrici di confusione ed al calcolo delle misure derivanti da essa, questa tecnica consente di creare un albero che, basandosi sui valori degli attributi che ogni immobile possiede, evidenzia le regole con cui a ciascun edificio viene assegnata la *label*. Riportiamo, per ogni esperimento, i valori dell'albero con cui si estraggono le regole per capire come il classificatore assegna la *label* a ciascun edificio. Per poter leggere queste regole, occorre partire dalla radice dell'albero ed arrivare sino alle foglie, ovvero quei nodi da cui non vengono create ulteriori diramazioni.

5.4 Esplorazione della conoscenza estratta

L'ultima fase del framework F-SCAN consiste nell'esplorazione, tramite l'ausilio di grafici riassuntivi, della conoscenza estratta. I risultati della cross-validation confermano che la *Features Selection* migliora l'intero processo di clustering, pertanto i grafici faranno riferimento unicamente ai dati che sono stati trattati nell'esperimento $E18$. L'esplorazione della conoscenza comincia con il recuperare alcune informazioni utili dal dataset originale, proveniente direttamente dal catasto delle certificazioni energetiche della Regione Piemonte: per estrarre queste informazioni, sfruttiamo l'operazione di join tra il dataset contenente gli attributi mantenuti dal *Processo di ottimizzazione*, la classe energetica e l'etichetta di clustering ed il dataset originale. Tramite l'ID di ogni edificio, risaliamo alle seguenti informazioni:

- Edificio ristrutturato: è un attributo booleano che indica in maniera sintetica se l'edificio è stato ristrutturato (SI) oppure no (NO). Le ristrutturazioni

presenti nel dataset sono segnalate solamente se queste sono state terminate dopo l'anno 2000;

- Anno costruzione: questo attributo indica l'anno in cui sono stati conclusi i lavori di costruzione dell'edificio;
- Provincia: è un attributo che indica la provincia in cui è ubicato l'immobile a cui la certificazione fa riferimento.

Questi tre attributi, confrontati con il cluster a cui l'edificio è stato assegnato e alla sua classe energetica, possono fornire importanti spunti per alcune analisi incrociate. Innanzitutto, nell'esperimento che consideriamo, l'algoritmo di clustering ha partizionato il dataset creando i seguenti cluster:

- cluster 0, avente edifici con performance energetica medio-alta;
- cluster 1, avente edifici con performance energetica alta;
- cluster 2, 5, aventi edifici con performance energetica media;
- cluster 3,4 e 6 aventi edifici con performance energetica bassa.

Consideriamo le ristrutturazioni. Prestiamo attenzione ai seguenti diagrammi a torta.

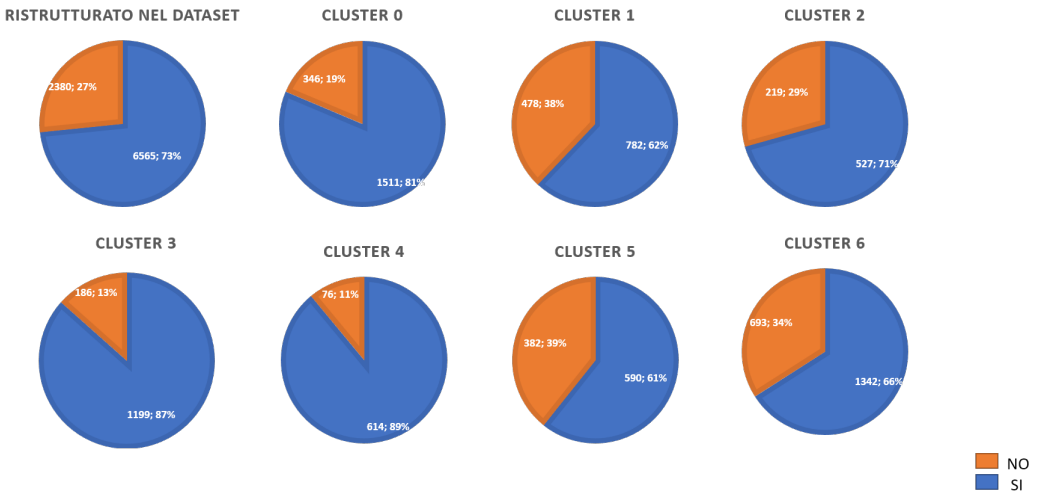


Figura 5.22: Edifici ristrutturati presenti nel dataset ed in ogni cluster

Il primo diagramma riassume quanti sono gli edifici che sono stati ristrutturati dopo il 2000 all'interno del dataset: solamente il 27%, ovvero 2380 edifici. Siamo facilmente portati a pensare che di questi 2380 palazzi, la maggior parte siano presenti nel cluster 1: è verosimile infatti che le ristrutturazioni portino delle migliorie che

aiutino ad incrementare l'efficienza energetica di ciascun edificio. In realtà, solamente 478 edifici, cioè il 21% circa di quelli ristrutturati, sono presenti nel cluster 1. Gli altri sono distribuiti quasi uniformemente negli altri cluster, ma la cosa sorprendente è che la maggior parte delle ristrutturazioni sono presenti nel cluster 6, quello con una performance energetica bassa. Alla luce di questa analisi, due sono le principali supposizioni che possiamo fare:

1. Sono state segnalate dal certificatore anche le piccole ristrutturazioni, ossia piccoli lavori edilizi che non sono andati a modificare la performance energetica;

2. I materiali e le tecniche costruttive, impiegati dalle società edilizie che hanno condotto i lavori, sono stati selezionati con logiche differenti rispetto a quella che prevede il miglioramento dell'efficienza energetica con conseguente riduzione dei consumi.

Riguardo al primo punto, possiamo suggerire ai certificatori di classificare le ristrutturazioni in piccole, medie e grandi a seconda delle modifiche che esse apportano all'edificio; con riferimento al secondo punto, occorrerebbe incentivare la scelta di tecniche e materiali che siano utili al miglioramento della performance energetica piuttosto che logiche quali il risparmio di costi su materiali e tecniche stesse.
Concentriamoci ora sulla figura 5.23. Essa è composta da istogrammi in cui,per ogni cluster, si riporta il periodo di costruzione degli edifici ristrutturati.
É naturale pensare che le costruzioni più datate siano quelle maggiormente soggette a ristrutturazione. In realtà, tutti gli istogrammi confermano che gli edifici soggetti a ristrutturazioni sono soprattutto quelli costruiti nel periodo 1951-1989. Sarebbe necessario incentivare la ristrutturazione degli immobili costruiti in periodi antecedenti rispetto a quello appena citato, perché concepiti con tecniche e materiali costruttivi più arretrate e quindi verosimilmente meno efficienti per quanto riguarda le prestazioni energetiche. Infatti, la figura 5.24 conferma ciò che abbiamo appena supposto.
La maggioranza degli edifici costruiti dopo il 2000 [3] sono collocati tutti nel cluster ad alta efficienza energetica, mentre gli immobili edificati prima del 1920 (sono inclusi anche palazzi medievali) si trovano soprattutto nel cluster con bassa performance energetica. I cluster a media e medio-bassa prestazione energetica possiedono gran parte degli edifici costruiti nel periodo 1950-1989. Pertanto, ribadiamo che sarebbe bene dare precedenza agli edifici più vecchi, perché le evidenze empiriche dimostrano che essi sono i meno performanti.
L'esplorazione della conoscenza estratta prosegue con l'analisi degli edifici per provincia. In figura 5.25 suddividiamo gli immobili a seconda della provincia in cui

[3] Ricordiamo che con "dopo il 2000" intendiamo fino al 2013, anno in cui è stato estratto il dataset di partenza

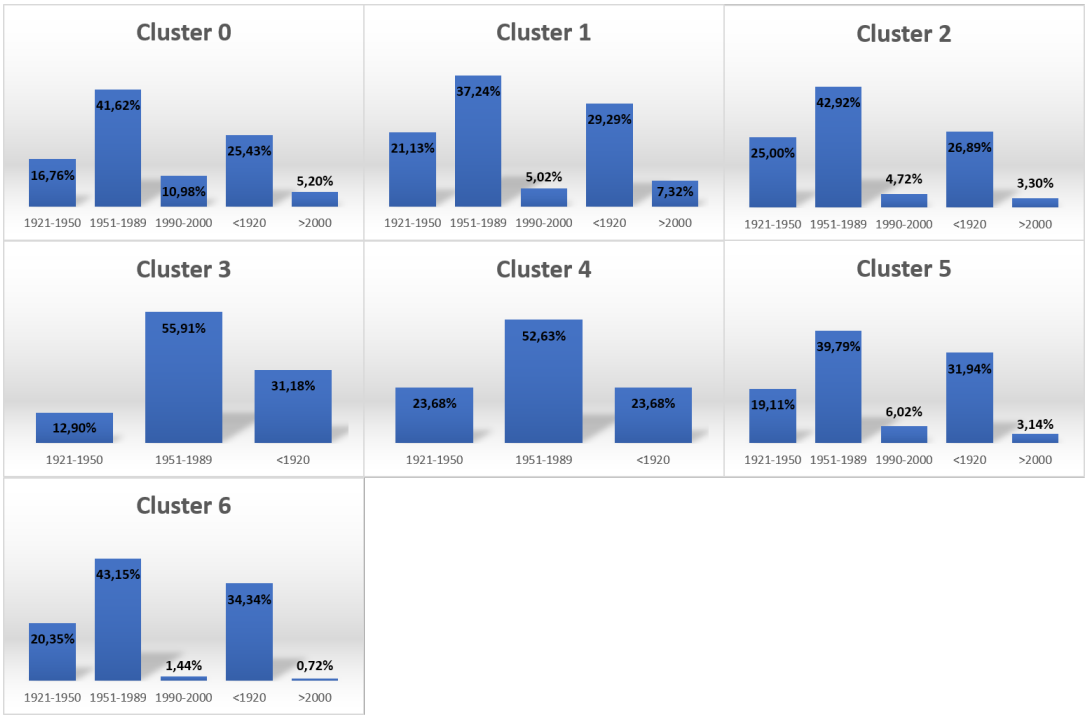


Figura 5.23: Periodo costruzione degli edifici ristrutturati

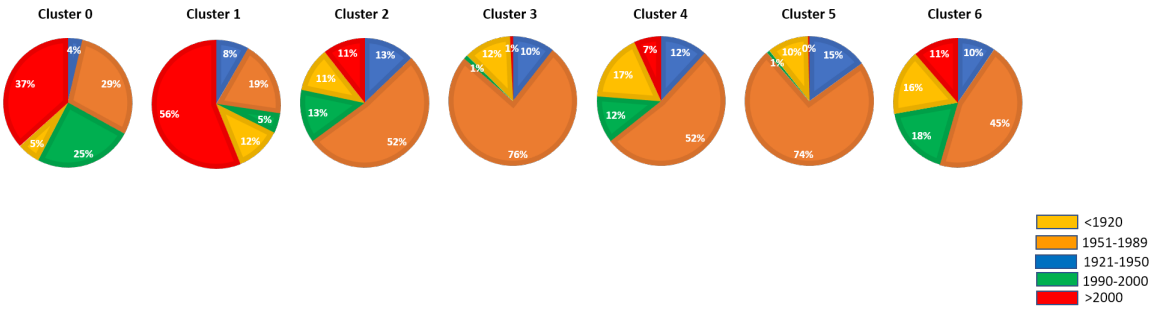


Figura 5.24: Periodo costruzione degli edifici ristrutturati

sono ubicati. Quasi metà delle certificazioni presenti nel nostro dataset si riferiscono ad immobili situati nella provincia di Torino. Come detto nel capitolo 1, le certificazioni energetiche sono divenute obbligatorie per gli edifici di nuova costruzione e per quelli soggetti a locazione. Torino, essendo una città ospitante più di 100.000 studenti, potrebbe essere in testa per numero di certificati emessi soprattutto per motivi legati alla locazione (l'Università di Torino ha sedi anche in altre province, ma il numero di studenti è molto minore rispetto alla sede di Torino). Fatte queste premesse, andiamo a vedere in figura 5.26 come si distribuiscono nei cluster le prestazioni energetiche degli edifici nelle differenti province. Utilizzeremo un *ma-*

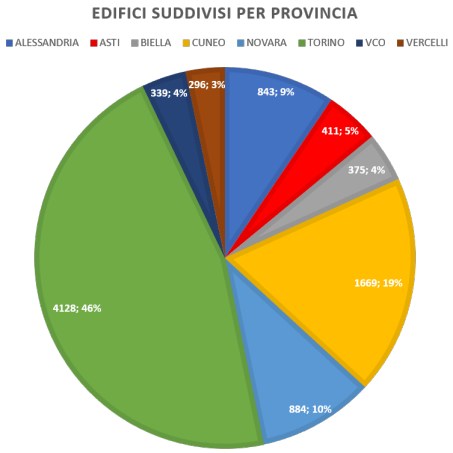


Figura 5.25: Raggruppamento per provincia degli edifici

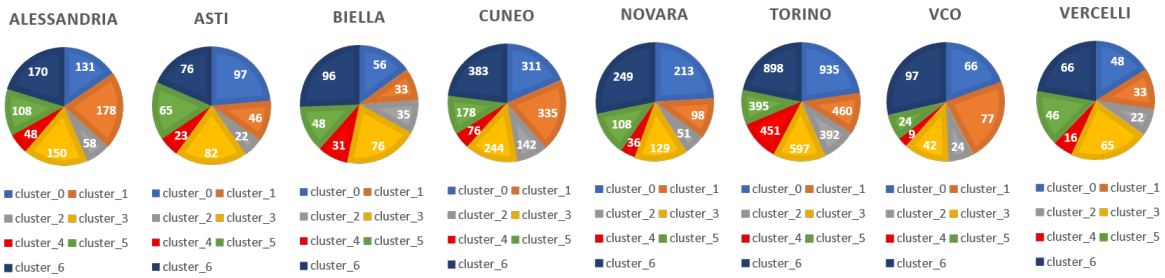


Figura 5.26: Suddivisione dei palazzi, raggruppati per provincia, nei cluster

jority model: per classificare una provincia in base alla sua prestazione, vedremo dove si collocano la maggior parte dei suoi edifici. Occorre tenere presente che la maggioranza degli immobili possiede un'efficienza energetica media. Dai diagrammi a torta, possiamo verificare che la provincia di Alessandria è quella che possiede più edifici con performance energetica alta, in quanto sono presenti nel cluster 1; gli immobili ubicati in provincia di Verbano-Cusio-Ossola sono quelli con efficienza energetica minore; i palazzi delle province di Torino e Asti hanno delle prestazioni medie e infine le province di Novara, Vercelli, Biella e Cuneo possiedono edifici con efficienza energetica medio-bassa. Non abbiamo informazioni a sufficienza per individuare le motivazioni principali del perché i palazzi residenziali di Alessandria consumino meno rispetto a quelli della provincia di VCO: potremmo ipotizzare che la provincia di Alessandria abbia avuto accesso a maggiori fondi provenienti dal Fondo nazionale efficienza energetica, ma non abbiamo trovato nulla a supporto di questa tesi. Riportiamo in figura 5.27 una cartina del Piemonte che riassume quanto appena detto.

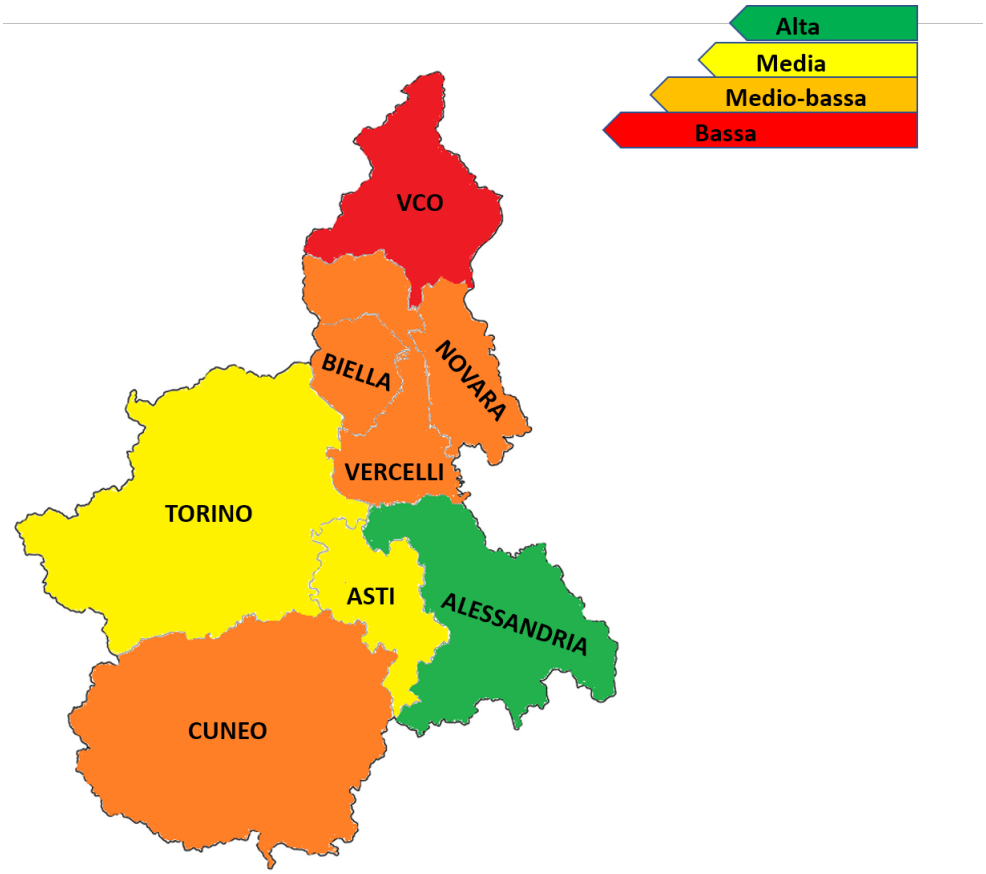


Figura 5.27: Performance energetica delle provincie piemontesi

Infine, la conoscenza estratta può essere esplorata mediante l'output del classificatore ad albero. Ricordiamo che la cross-validation da noi effettuata produce una matrice di confusione ed un albero di decisione. Percorrendo il *Decision Tree* dalla radice fino alle foglie, possiamo estrarre le regole con cui il classificatore assegna ogni palazzo ai vari cluster. L'albero che è stato creato dal processo di Rapid Miner è molto ampio e la sua lettura per intero poco agevole. Riportiamo di seguito un ramo significativo, ricavato dalla potatura dell'albero. In figura 5.28 vediamo che il classificatore inserisce nel cluster 1 (prestazione energetica alta) i palazzi che possiedono valori di trasmittanze opache minori di 0.271 e valori del fattore forma inferiori a 0.752. Nel caso in cui le trasmittanze opache siano minori di 0.271 ma il fattore forma ha un valore superiore a quello di soglia, il palazzo viene inserito nel cluster 2 (performance energetica media). Si rimanda all'apposita sezione dell'Appendice per la visualizzazione dell'albero per intero.

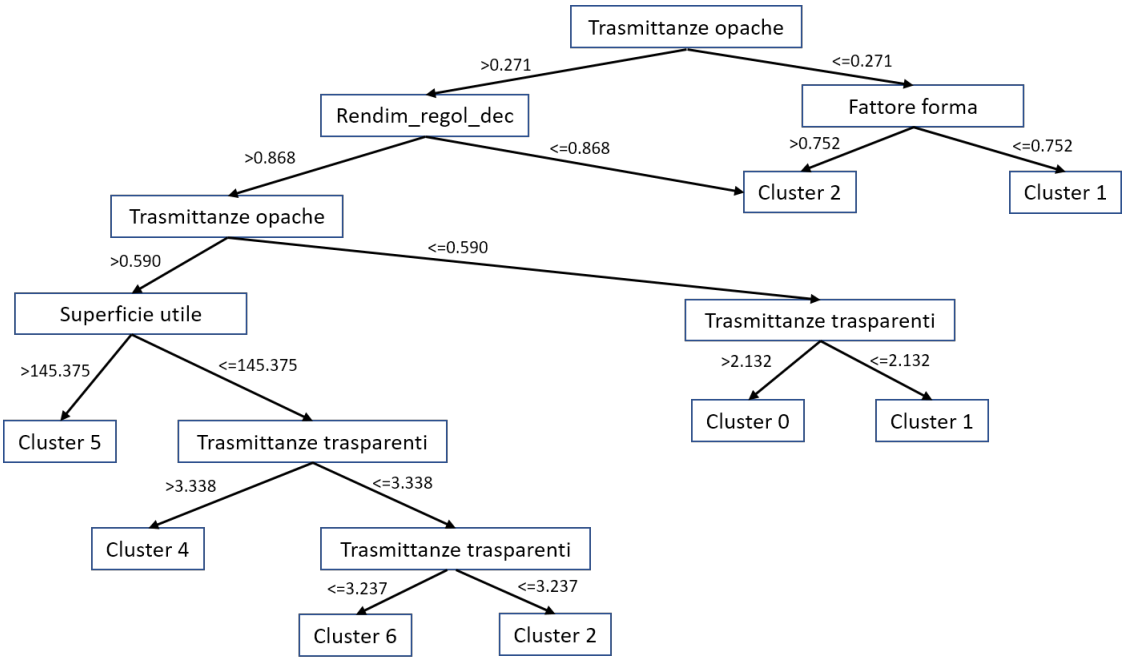


Figura 5.28: Un ramo del Decision Tree. Notiamo le regole con cui il classificatore assegna ciascun palazzo al cluster

Conclusioni e sviluppi futuri

L'obiettivo della tesi è stato l'implementazione di un'architettura utile per eseguire la profilazione degli edifici, con lo scopo di supportare le decisioni nell'ambito del processo di validazione delle certificazioni energetiche rilasciate e per contribuire alla creazione di un tool che possa automatizzare la procedura che porta alla concessione della Certificazione Energetica. In questo contesto è stato concepito il framework *F-SCAN*, un tool che implementa tecniche statistiche, di Data Mining e di cross-validation per profilare gli edifici a partire dai dati raccolti dal catasto delle certificazioni energetiche della Regione Piemonte.

Grazie all'*F-SCAN* è possibile individuare automaticamente gli attestati che non sono conformi ai limiti normativi in vigore, in quanto questi vengono riconosciuti come outliers. É altresì possibile individuare in automatico le caratteristiche termofisiche che determinano la performance energetica dell'edificio, trascurando le caratteristiche inutili, riuscendo a svincolare gli addetti dell'ufficio regionale di competenza dall'analisi direttamente le certificazioni dubbie anziché effettuare dei campionamenti.

Inoltre, grazie all'*F-SCAN* si riesce a monitorare l'evoluzione delle performances energetiche in funzione delle tecniche di costruzione che vengono stabilite dalle normative. I dataset su cui è stato applicato il framework sono composti da Certificazioni energetiche che sono state redatte secondo delle normative che sono state modificate; ciò non toglie che un approccio di questo tipo possa essere facilmente adattato anche alle nuove certificazioni. Se lo studio effettuato ha considerato solamente edifici residenziali, una sua possibile estensione potrebbe prevedere l'inclusione degli edifici adibiti ad altre destinazioni d'uso.

Un'ulteriore estensione potrebbe prevedere l'inclusione di altri fattori che determinano l'efficienza energetica degli edifici, per esempio dati riguardanti l'utilizzo di fonti di energia pulita come quella solare o eolica. Si potrebbe verificare il risparmio energetico percentuale che si otterrebbe se venisse ristrutturato l'edificio utilizzando altri tipi di materiali isolanti.

Si potrebbe pensare di impiegare dashboard dinamiche che guidano i certificatori nell'analisi dei dati in maniera tale da migliorare ed ottimizzare i tempi dell'intero processo.

Ma soprattutto è necessario sensibilizzare gli utenti al partecipare attivamente alla gestione efficiente delle risorse: si potrebbero introdurre incentivi a chi effettua una ristrutturazione dell'edificio con materiali e tecniche che ottimizzano il risparmio energetico; a chi effettua la sostituzione o il rinnovamento degli impianti di riscaldamento ed a chi utilizza in grossa percentuale energia proveniente da fonti rinnovabili.

Appendice A

Codice R completo

```
#Carico i packages necessari
requiredPackages <- c("broom", "knitr", "WriteXLS","dbscan", "grid", "readxl",
"nortest","factoextra","ngram", "jpeg", "sqldf","data.table","NbClust",
"matrixStats", "mctest","plyr")
ipak <- function(pkg){
new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]
if (length(new.pkg))
install.packages(new.pkg, dependencies = TRUE)
sapply(pkg, require, character.only = TRUE)
}
ipak(requiredPackages)

#Carico i dati
DataNorm <- read_xlsx("C:/Users/mirko/Desktop/TESI/TESI MIRKO/FILE 16
ATTRIBUTI/Dbscan Secondo Run.xlsx")
DataRange <- read_xlsx("C:/Users/mirko/Desktop/TESI/TESI MIRKO/FILE 16 ATTRIBUTI
/KMEANS/Dataset KMEANS.xlsx")

# Ciclo: Calcolo KNNdistance, plotto il grafico e lo salvo
for (i in c(1:20)){

kNNdistplot(DataRange ,i)
axis(2, at = seq(0,80,by = 1))
a=seq(0,80,by=1)
grid(nx = NULL, ny = NULL, col = "red", lty = "dotted")
abline(h=a,v=NULL,col="red",lty=1)
filename_k=concatenate(i,"NNdistance")
dev.copy(jpeg,filename=concatenate(filename_k,".jpeg"));
dev.off ();

}
```

```
KNN <- kNNdist(DataRange, 5) #calcolo kNNdist
KNN <- as.data.frame(KNN) #creo un data frame per poter modellare le colonne
#ordino il data frame in ordine decrescente sull'ultima colonna
KNN <- KNN[order(KNN$`5`, decreasing=TRUE), ]
a <- KNN$`5`#assegno i valori dell'ultima colonna ad una variabile chiamata "a"
plot(a) #plotto "a" per avere nuovamente il kNNdist
axis(2, at = seq(0,80,by = 2))
axis(1, at = seq(0,10000,by = 250))
asd1=seq(0,10000,by =250)
grid(nx = NULL, ny = NULL, col = "red", lty = "dotted")
abline(h=asd1,v=NULL,col="red",lty=250)
asd2=seq(0,80,by =2)
grid(nx = NULL, ny = NULL, col = "red", lty = "dotted")
abline(h=asd2,v=NULL,col="red",lty=2)
#creo due vettori di lunghezza pari a quella di a
secondDer<-vector(mode = "numeric",length=9105)
for(i in 2:(length(a)-1)){
 #assegno le derivate seconde agli elementi i del vettore "first"
secondDer[i] <- a[i+1]+a[i-1]-2*a[i]
#prendo leprime tre cifre dopo la virgola
secondDer[i] <- round(secondDer[i], digits = 3)

}#il ciclo calcola per ogni punto di "a" il valore della derivata seconda
derdiff <- diff(secondDer)
d <- order(derdiff, decreasing = TRUE)
primi50derdiff <- vector()
for(i in c(1:50)){
primi50derdiff[i] <- d[i]
}
View(primi50derdiff)
View(a)

#Carico il file originale con l'eticetta di cluster
DataLabelCluster <- read_xlsx("C:/Users/mirko/Desktop/TESI/
TESI MIRKO/FILE 16/DBSCAN.xlsx")
DataLabelCluster$cluster[DataLabelCluster$cluster=="cluster_0"] <- 0
DataLabelCluster$cluster[DataLabelCluster$cluster=="cluster_1"] <- 1
DataLabelCluster$cluster[DataLabelCluster$cluster=="cluster_2"] <- 2
DataLabelCluster$cluster[DataLabelCluster$cluster=="cluster_3"] <- 3
DataLabelCluster$cluster[DataLabelCluster$cluster=="cluster_4.0"] <- 4
DataLabelCluster$cluster[DataLabelCluster$cluster=="cluster_5"] <- 5
DataLabelCluster$cluster[DataLabelCluster$cluster=="cluster_6.0"] <- 6
cl <- sapply(DataLabelCluster$cluster,as.numeric)
```

```
 DataLabelCluster$cluster <- cl


 #creo e salvo i boxplot in un file pdf
 pdf("C:/Users/mirko/Desktop/TESI/TESI MIRKO/FILE 16 ATTRIBUTI/boxplots.pdf")
 b1 <- boxplot(DataLabelCluster$SUPERFICIE_UTILE~DataLabelCluster$cluster)
 b2 <- boxplot(DataLabelCluster$Altezza_Media~DataLabelCluster$cluster)
 b3 <- boxplot(DataLabelCluster$FATTORE_FORMA~DataLabelCluster$cluster)
 b4 <- boxplot(DataLabelCluster$TRASM_OPACHE~DataLabelCluster$cluster)
 b5 <- boxplot(DataLabelCluster$TRASM_TRASP~DataLabelCluster$cluster)
 b6 <- boxplot(DataLabelCluster$Rendim_Gener_Dec~
 DataLabelCluster$cluster)
 b7 <- boxplot(DataLabelCluster$Rendim_Regol_Dec~
 DataLabelCluster$cluster)
 b8 <- boxplot(DataLabelCluster$Rendim_Emiss_Dec~
 DataLabelCluster$cluster)
 b9 <- boxplot(DataLabelCluster$Rendim_Distr_Dec~
 DataLabelCluster$cluster)
 b10 <- boxplot(DataLabelCluster$EtaG_riscaldamento_TO~
 DataLabelCluster$cluster)
 b11 <- boxplot(DataLabelCluster$FABB_ENERGIA_TERMICA_UTILE_TO~
 DataLabelCluster$cluster)
 b12 <- boxplot(DataLabelCluster$INDICEPRESTAZIONEENERGACSTO~
 DataLabelCluster$cluster)
 b13 <- boxplot(DataLabelCluster$POT_RISCALDAMENTO~
 DataLabelCluster$cluster)
 b14 <- boxplot(DataLabelCluster$REND_MEDIO_GLOBSTAG_ETAGACS~
 DataLabelCluster$cluster)
 b15 <- boxplot(DataLabelCluster$INDICEPRESTAZEENERGACSTORINNO~
 DataLabelCluster$cluster)
 b16 <- boxplot(DataLabelCluster$RENDIM_STAGION_ACS_TO~
 DataLabelCluster$cluster)
 dev.off()

#carico il file con label cluster e classe energetica,
eseguo il raggruppamento per cluster e per classe
DataClassEnerg <- read_xlsx("C:/Users/mirko/Desktop/TESI/
TESI MIRKO/FILE 16 ATTRIBUTI/Kmeans .xlsx")
sqldf("select cluster,CLASSE_ENERG COUNT(*) FROM DataClassEnerg
 GROUP BY clster, CLASSE_ENERG")

DataKMeansZ <- read_xlsx("C:/Users/mirko/Desktop/TESI/TESI MIRKO/
FILE 16 ATTRIBUTI/Z/KMEANS.xlsx")
DataKMeansRange <- read_xlsx("C:/Users/mirko/Desktop/TESI/TESI MIRKO/
```

```
FILE 16 ATTRIBUTI/Dataset cluster.xlsx")
 # Elbow
 fviz_nbclust(DataKMeansRange, kmeans, method = "wss") +
 labs(subtitle = "Elbow method")
 # Silhouette method
 fviz_nbclust(DataKMeansRange, kmeans, method = "silhouette")+
 labs(subtitle = "Silhouette method")
 # NbClust function
 nb <- NbClust(DataKMeansRange, distance = "euclidean", min.nc = 2, max.nc = 10,
 method = "kmeans", index = "sdbw")
 fviz_nbclust(nb)

 #Analisi Descrittiva
 FileAnaDescr <- read_xlsx("C:/Users/mirko/Desktop/TESI/
 R/FILE 16 ATTRIBUTI/Cluster.xlsx")
 pdf("C:/Users/mirko/Desktop/TESI/R/FILE 16 ATTRIBUTI/istogrammi.pdf")
 hist(FileAnaDescr$Altezza_Media,main="Altezza media")
 hist(FileAnaDescr$TRASM_TRASP,main="Trasmittanze trasparenti")
 hist(FileAnaDescr$TRASM_OPACHE,main="Trasmittanze opache")
 hist(FileAnaDescr$FATTORE_FORMA,main="Fattore forma")
 hist(FileAnaDescr$Rendim_Gener_Dec,main="Rendimento di generazione")
 hist(FileAnaDescr$Rendim_Regol_Dec,main="Rendimento di regolazione")
 hist(FileAnaDescr$Rendim_Emiss_Dec,main="Rendimento di emissione")
 hist(FileAnaDescr$Rendim_Distr_Dec,main="Rendimento di distribuzione")
 hist(FileAnaDescr$EtaG_riscaldamento_TO,main="Rendimento globale riscaldamento")

 dev.off()

 #Regressione Lineare Multipla
 DatasetMLR <- read_xlsx("C:/Users/mirko/Desktop/TESI/TESI MIRKO/
 FILE 16 ATTRIBUTI/Dataset.xlsx")
 DatasetMLR <- DatasetMLR[-18]#tolgo ID
 #Trasformo la classe energetica da carattere a numerica
 DatasetMLR$CLASSE_ENERGETICA[DatasetMLR$CLASSE_ENERGETICA=="A+"] <- 8
 DatasetMLR$CLASSE_ENERGETICA[DatasetMLR$CLASSE_ENERGETICA=="A"] <- 7
 DatasetMLR$CLASSE_ENERGETICA[DatasetMLR$CLASSE_ENERGETICA=="B"] <- 6
 DatasetMLR$CLASSE_ENERGETICA[DatasetMLR$CLASSE_ENERGETICA=="C"] <- 5
 DatasetMLR$CLASSE_ENERGETICA[DatasetMLR$CLASSE_ENERGETICA=="D"] <- 4
 DatasetMLR$CLASSE_ENERGETICA[DatasetMLR$CLASSE_ENERGETICA=="E"] <- 3
 DatasetMLR$CLASSE_ENERGETICA[DatasetMLR$CLASSE_ENERGETICA=="F"] <- 2
 DatasetMLR$CLASSE_ENERGETICA[DatasetMLR$CLASSE_ENERGETICA=="G"] <- 1
 ClassEnerg <- sapply(DatasetMLR$CLASSE_ENERGETICA,as.numeric)
 DatasetMLR$CLASSE_ENERGETICA <- ClassEnerg
```

```
#Scelgo la variabile dipendente Y ed i regressori Xi
Y<- DatasetMLR$CLASSE_ENERGETICA
X1<- DatasetMLR$SUPERFICIE_UTILE
X2<- DatasetMLR$VOL_LORDO_RISCALDATO
X3<- DatasetMLR$Altezza_Media
X4<- DatasetMLR$SUP_DISPERDENTE_TOT
X5<- DatasetMLR$FATTORE_FORMA
X6<- DatasetMLR$TRASM_OPACHE
X7<- DatasetMLR$TRASM_TRASP
X8<- DatasetMLR$FABBENERGIA_TERMICA_UTILE_QH
X9<- DatasetMLR$‘FABBENERGIA_TERMICA_UTILE_ACS (kWh/mq)‘
X10<- DatasetMLR$ETA_acs
X11<- DatasetMLR$PRESTAZ_ENERG_ACS_EPACS_check
X12<- DatasetMLR$Rendim_Gener_Dec
X13<- DatasetMLR$Rendim_Regol_Dec
X14<- DatasetMLR$Rendim_Emiss_Dec
X15<- DatasetMLR$Rendim_Distr_Dec
X16<- DatasetMLR$RendimentoMedioGlobale_Check
X17<- DatasetMLR$EtaG_riscaldamento_TO
X18<- DatasetMLR$‘EtaG_riscaldamento+acs_TO‘
X19<- DatasetMLR$FABB_ENERGIA_TERMICA_UTILE_TO
X20<- DatasetMLR$INDICE_PRESTAZIONE_RISCALD_TO
X21<- DatasetMLR$INDICEPRESTAZIONEENERGACSTO
X22<- DatasetMLR$INDICE_PREST_ENERG_GLOBALE_TO
X23<- DatasetMLR$POT_RISCALDAMENTO
X24<- DatasetMLR$FABB_ACS_FONTIRINNO
X25<- DatasetMLR$REND_MEDIO_GLOBSTAG_ETAGACS
X26<- DatasetMLR$PRESTAZ_RAGGIUNGIB

#Analisi collinearità
DataCollAn <- DatasetMLR[,-27]#Escludo la response
DataCollAn <- as.matrix(DatasetMLR)
#calcola la varianza delle colonne
VarianzaAttributi <- colSds(DatasetCollAnalysis)
#matrice correlazione
CoMatrix <- cor(DatasetCollAnalysis, method = "pearson")
Cholesky <- chol(CorrelationMatrix) #metodo Cholesky
XMctest <- DatasetCollAnalysis
XMctest <- as.matrix(XMctest)

#funzione che permette di visualizzare complessivamente se c’è MC
omc <- omcdiag(XMctest,Y)
omc
#imcdiag fa un’analisi più approfondita
```

```
imc <- imcdiag(XMctest, Y)
imc
#Verifico che le features seguano una normale
DatasetMLR <- as.data.frame(DatasetMLR)
residuals <- regression$residuals
for(i in c(1:26)){
qqnorm(residuals, main = c(i,"QQ-PLOT"),
xlab = "Theoretical Quantiles", ylab = "Sample Quantiles",
plot.it = TRUE, datax = FALSE)

qqline(residuals, datax = FALSE, distribution = qnorm,
probs = c(0.25, 0.75), qtype = 7)
filename_k=concatenate("residuals",i,"QQ-Plot")
dev.copy(jpeg,filename=concatenate(filename_k,".jpeg"));
dev.off ()
}
plot(residuals,xlab = "Y Estimated")

#Regressione
View(DatasetMLR)
#DatasetMLR <- DatasetMLR[,-14]
reg<-lm(Y~X1+X2+X3+X4+X5+X6+X7+X8+X9+X10+
X11+X12+X13+X14+X15+X16,DataCollAn)
plot(regression$residuals)
tabella <- tidy(anova)
tabella <- as.table(a)
write.csv(tabella, file = "C:/Users/mirko/Desktop/TESI/
TESI MIRKO/FILE 16 ATTRIBUTI/Riepilogo MC e MR CSV.csv")
anova <- anova(regression)
summary(regression)
#test di ANDERSON DARLING per la normalità dei residui
ad.test(regression$residuals)
```

Bibliografia

[1] A. Acquaviva, D. Apiletti, A. Attanasio, E. Baralis, F. Castagnetti,T. Cerquitelli, S. Chiusano, E. Macii, D. Martellacci, and E. Patti. Enhancing energy awareness through the analysis of thermal energy consumption. In EDBT/ICDT - Energy Data Management (EnDM) Workshop, Brussels, Belgium, 2015.

[2] A. Capozzoli, D. Grassi, M. S. Piscitelli, and G. Serale. Discovering knowledge from a residential building stock through data mining analysis for engineering sustainability. Energy Procedia, 2015.

[3] J. Han, J. Pei, and M. Kamber. Data mining: concepts and techniques. Elsevier, 2011.

[4] V. K. T. Pang-Ning, M. Steinbach. Introduction to Data Mining, (First Edition). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[5] S. Dulli, S.Furini, E.Peron. Data mining: metodi e strategie. Springer, 2009.

[6] A. Iacono. Caratterizzazione delle certificazioni energetiche mediante tecniche di data mining. Caso di studio: Regione Piemonte. Torino, 2016.

[7] J.Stock, M.Watson Introduzione all'econometria. Milano, Pearson Education, 2005.

[8] S.M.Ross. Introduzione alla statistica. Milano, Maggioli Editore, 2014.

[9] S.J. Julier, J.K. Uhlmann, A General Method for Approximating Nonlinear Tranformations of ProbabilityDistributions, 1997.

[10] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation andvalidation of cluster analysis. Journal of Computational and Applied Mathematics, 1987.

[11] A. Capozzoli, G. Serale, M. S. Piscitelli, and D. Grassi. Data mining for energy analysis of a large data set of flats. Proceedings of the institution of civil engineers. Engineering sustainability, 2017.

[12] H. Abdi, L.J. Williams. Principal component analysis, 2010.

[13] B.-H. Juang and L. Rabiner. The segmental k-means algorithm for estimating parameters of hidden markov models. IEEE Transactions on Acoustics, Speech and Signal Processing, 1990.

[14] S. M. Ross. Introduction to probability and statistics for engineers and scientists (2. ed.). Academic Press, 2000.

[15] I. Takouna, E. Alzaghoul, and C. Meinel. Robust virtual machine consolidation for efficient energy and performance in virtualized data centers. In IEEE iThings/GreenCom/CPSCom 2014, Taipei, Taiwan, September 1-3, 2014, pages 470–477, 2014.

[16] M. Ester, H. Kriegel, J. Sander, X. Xu. A Density-Based Algorithm for Discovering Clusters. University of Munich, 1996.

[17] D. Farrar, R. Glauber. Multicollinearity in Regression Analysis: The Problem Revisited in Large Spatial Databases with Noise. The MIT Press, 1967.

[18] T.Ozyer, R. Alhajj. Effective Clustering by Iterative Approach. University of Calgary, 2005.

[19] G. Golub. Matrix Decompositions and Statistical Calculations. New York, 1969.

[20] M. Charrad, N.Ghazzali, V. Boiteau, A. Niknafs. NbClust Package : finding the relevant number of clusters in a dataset. University of Quebec, 2012.

[21] I. Khan, A. Capozzoli, S. Corgnati, T. Cerquitelli. Fault Detection Analysis of Building Energy Consumption Using Data Mining Techniques. Torino, 2013.

[22] E. Di Corso, T. Cerquitelli. Characterizing thermal energy consumption through exploratory data mining algorithms. Torino, 2016.

[23] I. Guyon, A.Elisseeff. An Introduction to Variable and Feature Selection. Berkeley, 2003.

[24] B. Zheng, S. W. Yoon, and S. S. Lam. Breast cancer diagnosis based on feature extraction using a hybrid of k-means and support vector machine algorithms. Expert Systems with Applications, 2014.

[25] Regressione lineare, ANOVA, Decomposizione di Cholesky. Link consultati:
`https://it.wikipedia.org/wiki/Regressione_lineare`;
`https://it.wikipedia.org/wiki/Decomposizione_di_Cholesky`;
`https://it.wikipedia.org/wiki/Indice_di_correlazione_di_Pearson`;
`https://it.wikipedia.org/wiki/Analisi_della_varianza`.
Ultimo accesso: luglio 2018.

[26] T. Kodinariya, P. Makwana .Review on determining number of Cluster in K-Means Clustering. India, 2013.

[27] J. Hartigan. Clustering algorithms. New York, 1975