POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Gestionale

Tesi di Laurea Magistrale

Caratterizzazione e visualizzazione delle prestazioni energetiche degli edifici a partire da dati open.

Caso di studio: la città di Torino

Relatori:
prof. Tania Cerquitelli
prof. Elena Maria Baralis

Candidato:
Fabio Bellotti

Tutore Aziendale
Edison Spa
Ing. Silvia Casagrande

Dicembre 2018

Ringraziamenti

Innanzitutto desidero ringraziare la Prof.ssa Tania Cerquitelli e la Prof.ssa Elena Maria Baralis, per avermi permesso di lavorare a questo progetto di tesi. Desidero ringraziare in particolare la Prof.ssa Tania Cerquitelli per l'aiuto fornitomi, la competenza e la cortesia.

Si ringrazia la Edison Spa, in particolare l'Ing. Silvia Casagrande, per la disponibilità, la gentilezza e la grande professionalità. Si ringraziano inoltre i colleghi delle Officine Edison, per la calorosa accoglienza, l'interesse e il supporto.

Il presente lavoro di tesi è stato svolto in collaborazione con il Dipartimento Energia del Politecnico di Torino. Pertanto, desidero ringraziare l'Ing. Alfonso Capozzoli, per le nozioni, la competenza dimostrata e i preziosi consigli.

Inoltre, ringrazio la Dott.ssa Evelina Di Corso e il Dott. Stefano Proto per il tempo dedicatomi, la pazienza, e la professionalità.

Uno speciale ringraziamento va ai miei genitori, per aver ascoltato i miei infiniti discorsi e per aver sempre creduto in me, a mia sorella per avermi sempre strappato un sorriso, a mia zia Lory e i miei nonni per esserci sempre, nei momenti più importati.

Voglio ringraziare Raluca, per essermi sempre stata vicina, per l'affetto e la comprensione.

Desidero ringraziare gli amici di sempre, per il supporto, le risate e i discorsi fino a tarda notte.

Infine, un ringraziamento va ai miei colleghi, senza i quali il percorso sarebbe stato di sicuro più noioso e difficile, e in particolare a Maria Giovanna, con la quale ho condiviso questi mesi di lavoro.

Indice

Elenco delle figure

Elenco delle tabelle

Introduzione

Lo studio di soluzioni e modelli che migliorino l'efficienza delle nostre città, è sempre più un tema fondamentale per quanto riguarda il controllo dell'inquinamento del nostro pianeta; più della metà della popolazione mondiale risiede in aree urbane, e si stima che questo numero aumenterà fino al 69% (7.1 miliardi di persone) entro il 2050 [1]. Tali aree urbane necessitano dei due-terzi della domanda di energia primaria, e sono responsabili del 70% del totale di emissioni di anidride carbonica. Gli edifici residenziali sono responsabili di quasi il 60% del consumo di energia per gli edifici, e in Europa è richiesta una riduzione di emissioni di circa il 90% entro il 2050. Inoltre, l'introduzione delle nuove normative in Italia per gli APE (*Attestazione di Prestazione Energetica*), garantisce un formato standard e centralizzato di archiviazione dell'informazione nel *SIAPE* (*Sistema Informativo sugli Attestati di Prestazione Energetica*), relativa a caratteristiche riguardanti il fabbisogno energetico, le prestazioni, e le proprietà termofisiche di un edificio. In questo scenario, la disponibilità sempre più libera di grandi quantità di dati, permette l'utilizzo di tecniche di *data mining* come strumento atto ad estrarre conoscenza utile e a creare modelli decisionali[2]. Lo studio effettuato in questo lavoro di tesi, si pone l'obiettivo di caratterizzare gli edifici della città di Torino dal punto di vista dell'efficienza energetica, facendo uso di tecniche di *data mining* applicate sugli attestati di prestazione energetica relativi agli anni dal 2016 al primo semestre del 2018. Per raggiungere questo scopo, è stato sviluppato un *framework* che supporti l'analista durante l'estrazione e la visualizzazione della conoscenza. Il *framework* è stato sviluppato in *Python*, e si compone di due fasi principali:

- *(i)* partizionamento del *dataset* in gruppi omogenei di edifici, caratterizzati da proprietà dell'involucro, termofisiche, e di efficienza simili;
- *(ii)* visualizzazione della conoscenza estratta con l'ausilio dei grafici e mappe interattive, in grado di fornire informazioni relative agli attributi del *dataset* e alla conoscenza estratta da questi.

Le mappe sono state realizzate con l'intento di rendere fruibili e consultabili i risultati anche ai non addetti ai lavori, grazie alla possibilità visualizzare informazioni aggregate in modo intuitivo.

Il lavoro si articola in 5 capitoli.

- **Capitolo 1**: Viene presentata la normativa riguardo la certificazione energetica.
- **Capitolo 2**: Viene presentato il processo di estrazione della conoscenza.
- **Capitolo 3**: Viene presentato il *framework* utilizzato nell'analisi.
- **Capitolo 4**: Vengono presentati per ogni area del *framework* i risultati sperimentali ottenuti, sia in forma tabellare, che con l'ausilio di mappe e grafici.
- **Capitolo 5**: Vengono presentate le conclusioni del lavoro e gli eventuali sviluppi futuri.

Capitolo 1

La Certificazione Energetica

La Certificazione Energetica (o Certificato Energetico) è un documento che descrive le caratteristiche energetiche di un edifico, prendendo in esame le sue caratteristiche termofisiche e geometriche, oltre che gli aspetti caratteristici della zona climatica di appartenenza; inoltre fornisce una valutazione sintetica dell'efficienza energetica, assegnando un etichetta di classe da A4 a G. Questa procedura di valutazione è stata prevista dalle direttive europee 2002/91/CE[1] e 2006/32/CE[2], e si colloca all'interno del progetto iniziato con il protocollo di Kyoto nel 1997 in tema di tutela ambientale e riduzione dell'emissione di gas serra, volta a limitare il surriscaldamento globale. L'Europa si inserisce in questo quadro mettendo in primo piano l'importanza dell'efficientamento energetico delle sue città; come riportato da uno studio condotto dall'Istituto per le Energie Rinnovabili dell'EURAC (Accademia Europea di Bolzano), il 40% del consumo di energia europeo è da attribuirsi agli edifici, e circa i due terzi di questi sono causati dal riscaldamento[3]. In Italia sono stati introdotti provvedimenti su questo tema a partire dal 2005 tramite il decreto legislativo

[1] http://efficienzaenergetica.acs.enea.it/doc/dir91-02.pdf

[2] http://efficienzaenergetica.acs.enea.it/doc/dir32-06.pdf

[3] http://www.inspirefp7.eu/about-inspire/downloadable-reports/

192/2005[4] con l'intento di migliorare il rendimento energetico degli edifici in termini di efficienza energetica, grazie all'informazione fornita ai proprietari e utilizzatori, circa i consumi energetici richiesti per mantenere determinate condizioni ambientali interne. Negli anni, il decreto originale ha subito diversi aggiornamenti, fra i quali possiamo citare il 63/2013[5] dove viene introdotto il cosiddetto APE (*Attestato di Prestazione Energetica*) sostituendo il precedente ACE (*Attestato di Certificazione Energetica*), e successivamente i tre decreti ministeriali del 2015; tra le principali modifiche vengono determinate nuove metodologie di calcolo delle prestazioni e di assegnazioni delle relative classi energetiche, oltre che le linee guida per la compilazione della relazione tecnica. Il nuovo APE, come definito nel DM 26/06/2015[6], esprime la prestazione energetica globale in termini di energia primaria non rinnovabile. Questa variabile definita con la sigla $EP_{\text{gl,nren}}$ è espressa in $kWh/mq*anno$, e rappresenta l'energia che deve essere consumata affinché l'edificio (o l'unità immobiliare) raggiunga le condizioni di comfort; viene calcolata come la somma del fabbisogno di energia primaria non rinnovabile per la climatizzazione invernale ed estiva ($EP_{\text{H,nren}}$ ed $EP_{\text{C,nren}}$), de fabbisogno per la produzione di acqua calda sanitaria ($EP_{\text{W,nren}}$), per la ventilazione ($EP_{\text{V,nren}}$) e, nel caso del settore non residenziale, per l'illuminazione artificiale ($EP_{\text{L,nren}}$) e il trasporto di persone o cose ($EP_{\text{T,nren}}$).

$$EP_{gl,nren} = EP_{H,nren} + EP_{C,nren} + EP_{W,nren} + EP_{V,nren} + EP_{L,nren} + EP_{T,nren}$$

La determinazione finale della classe viene definita tramite un confronto tra l'indice di prestazione $EP_{\text{gl,nren}}$ dell'edificio con quello del suo rispettivo edificio di riferimento; questo rappresenta un'edificio identico al corrispettivo reale in termini di geometria, ubicazione territoriale e destinazione d'uso, avente caratteristiche termiche ed energetiche standard. A seguito di questo confronto verrà attribuita una etichetta di classe rappresentata da un'indice alfanumerico compreso tra **A4** e **G**, dove **A4** indica la classe più efficiente e **G** indica la peggiore. Va sottolineato come

[4] http://www.acs.enea.it/doc/dlgs_192-05.pdf

[5] http://www.gazzettaufficiale.it/eli/id/2013/06/05/13G00107/sg

[6] https://www.cisl.it/attachments/article/648/Decreto%20efficenza%20energetica%201.pdf

pur rispettando le direttive nazionali, spetti alle Regioni deliberare in merito alla raccolta e alla gestione dei dati dell'APE.

1.1 La Regione Piemonte

Il *dataset* analizzato contiene dati relativi alla regione Piemonte, memorizzati all'interno del Sistema informativo per la Prestazione Energetica degli Edifici (SIPEE). Questo è stato realizzato grazie al supporto del Consorzio per il Sistema Informativo (CSI-Piemonte), e gestisce l'elenco regionale dei soggetti abilitati al rilascio dell'APE, i dati inseriti nei certificati, e la loro raccolta dopo che vengono trasmessi dai certificatori professionisti. Il sistema risulta centralizzato, ed eliminando l'inserimento manuale dei dati negli attestati APE, limita inoltre gli errori umani di calcolo e trascrizione; i risultati intermedi e finali degli APE vengono infatti generati automaticamente da un limitato numero di software certificati ed esportati in formato XML (*eXtensible Markup Language*). Si ricorda che l'APE è obbligatorio, e deve essere presentato alla Pubblica Amministazione per ogni atto di compravendita e locazione di immobili, oltre che nel caso di ristrutturazioni che insistano su oltre il 25% della superficie dell'involucro (pareti e tetti) dell'intero edificio. In linea con i temi dell' *open data* e della *open innovation*, il CSI-Piemonte ci ha fornito il *dataset* di certificati raccolti dall'anno 2016.

1.2 Descrizione degli Attributi Principali

Di seguito verranno descritti gli attributi dell'APE più rilevanti ai fini della caratterizzazione energetica. I dati contenuti al suo interno fanno parte di diverse macro-aree:

- dati catastali

- dati tecnici generali sul fabbricato
- dati sui rendimenti
- dati sugli impianti

Per la nostra analisi ci siamo concentrati sulla porzione di *database* più densa per numero di certificati, filtrando quindi i certificati a nostra disposizione per la città di Torino e facenti parte della destinazione d'uso E1(1), ovvero abitazioni adibite a residenza con carattere continuativo. Tra questi si sono selezionati solo quelli relativi ad un'unità immobiliari. I vari attributi sono memorizzati dal SIPEE in *dataset* disgiunti. Si è reso quindi necessario come prima operazione effettuare una *join* dei diversi *dataset* in formato CSV (*Comma Separated Values*) su di un attributo o una serie di attributi che identifichino univocamente un singolo edificio o alloggio; per evitare di ottenere certificati multipli per uno stesso alloggio si sono scelte come chiavi della *join* tre attributi:

- **Foglio**: anche detto foglio catastale, identifica una porzione di territorio comunale.
- **Particella**: nell'ambito di un foglio catastale indica una porzione di terreno, o un fabbricato e l'eventuale area di pertinenza; viene contrassegnata da un numero ed è anche nota come mappale, o numero di mappa.
- **Subalterno**: nell'ambito del catasto fabbricati consente l'identificazione di un bene immobile, compresa la singola unità immobiliare esistente su di una particella.

Nel caso di certificati multipli per una stessa unità abitativa, si è deciso di selezionare il certificato con data di upload più recente.

1.2.1 Dati Catastali

Come anticipato, la prima tipologia di dati che andremo ad analizzare sono quelli di tipo catastale. Questi sono stati di fondamentale importanza nella fase iniziale per poter filtrare il *dataset*, e in fase di visualizzazione hanno permesso di localizzare su mappe l'informazione e la conoscenza ottenuta. Oltre alle informazioni fornite dagli attributi *Foglio*, *Particella* e *Subalterno*, sono stati presi in considerazione anche:

- **CAP**: numero di avviamento postale. Per la città di Torino nello specifico i valori del CAP sono compresi tra 10121 e 10155.
- **Provincia**: la provincia a cui si riferisce l'APE specifico, tutte facenti parte della regione Piemonte.
- **Comune**: il comune a cui si riferisce APE in questione, facente parte di una specifica provincia.
- **Numero Civico**: il numero civico dell'unità immobiliare in esame.
- **Indirizzo**: l'indirizzo dell'unità immobiliare oggetto dell'APE.
- **Latitudine**: la latitudine è la distanza angolare misurata in gradi lungo l'arco di meridiano compreso tra l'Equatore e il parallelo passante per il punto considerato.
- **Longitudine**: la longitudine è la distanza angolare misurata in gradi, lungo l'arco di parallelo compreso tra il Meridiano fondamentale (Meridiano di Greenwich) e il meridiano passante per il punto considerato.
- **Anno di Costruzione**: l'anno di costruzione dell'edificio.

Poiché i campi Indirizzo e Numero Civico presentano dei campi testuali liberi, questi hanno richiesto una fase di pulizia per controllarne la consistenza; l'operazione ha inoltre permesso di validare le informazioni di Latitudine e Longitudine ad essi associate.

1.2.2 Dati Tecnici Generali sul Fabbricato

I dati tecnici generali sul fabbricato riguardano informazioni sulla struttura dell'unità immobiliare da un punto di vista geometrico e fisico.

- **Destinazione d'uso**: per destinazione d'uso s'intendono l'insieme delle modalità e delle finalità di utilizzo di un immobile. Si riporta nella Tabella 1.1 il dettaglio sui possibili valori che può assumere l'attributo in esame.
- **Fattore forma** $[m^{-1}]$: viene definito con $\frac{S}{V}$, dove S indica la superficie disperdente mentre V si riferisce al volume lordo riscaldato dell'unità immobiliare[3]. Anche indicato come rapporto di compattezza (*compactness ratio*), dipende dalla forma dell'edificio e dalle sue dimensioni. Il fattore forma influenza l'entità delle dispersioni termiche per trasmissione oltre che il fabbisogno energetico per la sua climatizzazione invernale ed estiva. Ai fini della valutazione energetica di un edificio, o di un'unità immobiliare, questa è tanto più performante quanto è più compatto il suo involucro termico, a parità di altri parametri quali isolamento, impianti installati e orientamento.
- **Superficie utile** $[m^2]$: rappresenta la superficie netta calpestabile espressa in m^2. Questa tiene conto di gli ambienti climatizzati al netto di muri esterni e tramezzi, ed comprende le soglie delle porte e gli spazi al disotto dei terminali di emissione[3].

1.2.3 Dati Tecnici Specifici

In questa sezione verranno considerati i dati tecnici specifici dell'involucro e dell'impianto di climatizzazione, caratterizzanti la prestazione energetica presenti nell'APE. La trasmittanza termica U è il parametro principale utilizzato per calcolare le dispersioni termiche attraverso l'involucro di un edificio. Rappresenta il flusso di calore che attraversa una superficie unitaria sottoposta a differenza di temperatura pari ad $1°C$ e si misura in $\frac{W}{m^2K}$. La norma di riferimento per il calcolo della trasmittanza

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

Tabella 1.1: Categorie delle destinazioni d'uso degli edifici

termica è la UNI EN ISO 6946:2018. Più basso è il valore di trasmittanza termica degli elementi che costituiscono l'involucro edilizio, minore sarà il flusso di calore che attraversa gli elementi stessi. Un basso valore di U consente quindi di ridurre le dispersioni di calore e garantire una migliore efficienza dell'involucro. La trasmittanza si divide in:

- **Trasmittanza opaca** $[\frac{W}{m^2K}]$: misura le dispersioni termiche attraverso gli elementi opachi dell'edificio (muri perimetrali) verso l'ambiente esterno,ed è calcolata come una media ponderata dei vari apporti.
- **Trasmittanza trasparente** $[\frac{W}{m^2K}]$: misura le dispersioni termiche attraverso gli elementi trasparenti dell'edificio (infissi in genere) verso l'ambiente esterno, ed è calcolata come una media ponderata dei vari apporti.

Un'altra categoria di attributi riguarda i rendimenti dei sistemi di climatizzazione e di creazione di acqua calda sanitaria (ACS); è infatti noto come esistono perdite di efficienza nell'utilizzo reale degli impianti rispetto ai rendimenti nominali dichiarati dai costruttori. I rendimenti che prenderemo in considerazione sono:

- **Rendimento di generazione**: il rendimento di generazione medio stagionale $ETAG$ si riferisce al rendimento del sistema adibito alla produzione di acqua calda sanitaria e alla climatizzazione invernale. È calcolato come il rapporto fra il calore utile prodotto dal generatore nella stagione di riscaldamento e l'energia fornita nello stesso periodo sotto forma di combustibile ed energia elettrica. La perdita di efficienza rispetto alla condizione ideale è dovuta al fatto che non tutta l'energia fornita viene trasferita all'acqua.
- **Rendimento di emissione**: il rendimento di emissione medio stagionale $ETAE$ è definito come il rapporto fra il calore richiesto per il riscaldamento degli ambienti con un sistema di emissione teorico di riferimento in grado di fornire una temperatura ambiente perfettamente uniforme ed uguale nei vari locali ed il sistema di emissione reale, nelle stesse condizioni di temperatura ambiente e di temperatura esterna. Il rendimento di emissione è quasi sempre

inferiore all'unità, dal momento che i moti convettivi innescati dal sistema di emissione, soprattutto quando movimentino attivamente l'aria o irradino direttamente una parete disperdente, aumentano i coefficienti di scambio, ovvero le dispersioni.

- **Rendimento di regolazione**: il rendimento di regolazione medio stagionale *ETAR* è il rapporto tra il calore richiesto per il riscaldamento degli ambienti con una regolazione teorica perfetta e il calore richiesto per il riscaldamento degli stessi ambienti con un sistema di regolazione reale. Il regolatore teorico perfetto è quello in grado di ridurre immediatamente l'emissione del corpo scaldante in presenza di un apporto di calore proveniente da fonte diversa dall'impianto di riscaldamento. Il regolatore reale riduce l'emissione del corpo scaldante solo dopo che l'apporto gratuito ha provocato un aumento di temperatura, generando delle inefficienze.
- **Rendimento di distribuzione**: il rendimento di distribuzione medio stagionale *ETAD* è il rapporto fra la somma del calore utile emesso dai corpi scaldanti e del calore disperso dalla rete di distribuzione all'interno dell'involucro riscaldato dell'edificio ed il calore in uscita dall'impianto di produzione ed immesso nella rete di distribuzione. Il rendimento di distribuzione caratterizza l'influenza della rete di distribuzione sulla perdita passiva di energia termica, ovvero quella non ceduta agli ambienti da riscaldare.
- **ETAH**: calcolato come il prodotto dei quattro rendimenti *ETAG*, *ETAE*, *ETAR* ed *ETAD*, fornisce un'indicazione riassuntiva dell'efficienza di un determinato sistema di climatizzazione.

Per quanto riguarda gli indici di prestazione si sono considerati anche i seguenti attributi:

- $\mathbf{EP_{gl,nren}}$ $[\frac{kW}{m^2}]$: indice di prestazione energetica non rinnovabile globale dell'edificio.
- $\mathbf{EP_{H,nren}}$ $[\frac{kW}{m^2}]$: indice di prestazione energetica non rinnovabile per la climatizzazione invernale.

- **$EP_{C,nren}$** $[\frac{kW}{m^2}]$: indice di prestazione termica non rinnovabile utile per il raffrescamento.

- **$EP_{W,nren}$** $[\frac{kW}{m^2}]$: indice di prestazione termica non rinnovabile utile per la produzione di acqua calda sanitaria.

- **$EP_{V,nren}$** $[\frac{kW}{m^2}]$: indice di prestazione energetica non rinnovabile per la ventilazione.

- **$EP_{L,nren}$** $[\frac{kW}{m^2}]$: indice di prestazione energetica non rinnovabile per l'illuminazione artificiale.

- **$EP_{T,nren}$** $[\frac{kW}{m^2}]$: indice di prestazione energetica non rinnovabile del servizio per il trasporto di persone e cose.

  I contributi apportati da $EP_{L,nren}$ e $EP_{T,nren}$ al calcolo finale del rendimento globale $EP_{gl,nren}$ non vengono considerati nel caso di edifici residenziali.

- **$EP_{H,nd}$** $[\frac{kWh}{m^2}]$: indice di prestazione termica utile per riscaldamento . Si calcola come:

$$EP_{\mathrm{H,nd}} = \frac{Q_{\mathrm{H,nd}}}{S_{\mathrm{utile}}}$$

  dove:

  - **$Q_{H,nd}$** $[kWh]$: è il fabbisogno di energia termica utile ideale per il riscaldamento.
  - **S_{utile}** $[m^2]$: è la superficie utile dell'unità immobiliare.

  L'indice $EP_{H,nd}$ deve risultare inferiore al valore del corrispondente indice limite calcolato per l'edificio di riferimento ($EP_{H,nd,limite}$) per il quale i parametri energetici e le caratteristiche termiche sono dati nelle pertinenti tabelle dell'appendice A del decreto in questione per i corrispondenti anni di vigenza.

1.2.4 La Classe Energetica

La classe energetica è un'indicazione sintetica del grado di efficienza energetica di un edificio inteso come singola unità immobiliare o come intero edificio. A seguito della riforma del 2015, le classi energetiche sono passate da 7 a 10, come mostrato in Figura 1.1. La grande novità introdotta dalla riforma è l'introduzione nella metodologia di calcolo e assegnazione della classe energetica dell'*edificio di riferimento*. Questo rappresenta un edificio ideale avente caratteristiche termofisiche similari a quelle dell'edificio reale, e avente prestazioni energetiche standard; l'assegnazione della classe energetica sarà così riportata a questo riferimento, potendo risultare l'edifico reale uguale, meno o più efficiente dell'edificio di riferimento.

Figura 1.1: Rappresentazione delle classi energetiche in vigore dal 2015

Gli intervalli di prestazione che identificano le classi energetiche sono ricavati attraverso coefficienti moltiplicativi di riduzione/maggiorazione del valore $EP_{gl,nren}$, (2019/21).

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

Figura 1.2: Intervalli delle classi energetiche in vigore dal 2015

Capitolo 2

Estrazione della Conoscenza

La mole di dati memorizzata con l'ausilio di supporti informatici sta subendo una crescita continua; riuscire a sfruttare questa grandissima risorsa è una delle sfide cardine del nostro decennio. Grandi colossi mondiali come Amazon, Google ed Ebay fondano infatti il loro vantaggio competitivo proprio sulla loro capacità di sfruttare i *Big Data* per essere sempre più *responsive* ai cambiamenti del mercato e ai gusti del singolo cliente. In questo contesto sono invece ancora molte le aziende che pur producendo e memorizzando una grandissima quantità di dati, non adottano tecniche KDD per estrarre conoscenza utile da questi. Il termine *Knowledge Discovery in Databases* (KDD) è stato coniato nel 1989 per descivere la conoscenza come il prodotto finale di un processo di scoperta *data-driven* [4], che a partire dai dati grezzi porta ad un'informazione spendibile per il *decision-making*, costituita di *pattern*. Un *pattern* è un'espressione che descrive un subset dei dati esaminati o un modello applicato a questo sottoinsieme. Il processo KDD è interattivo ed iterativo, e si compone di diversi step:

1. Sviluppare una comprensione del dominio di appartenenza ed applicazione dei dati e identificare l'obiettivo del processo KDD.
2. Creare un *dataset target*, ovvero selezionare un *dataset*, o concentrarsi su di un sottoinsieme di variabili e dati su cui iniziare la procedura.

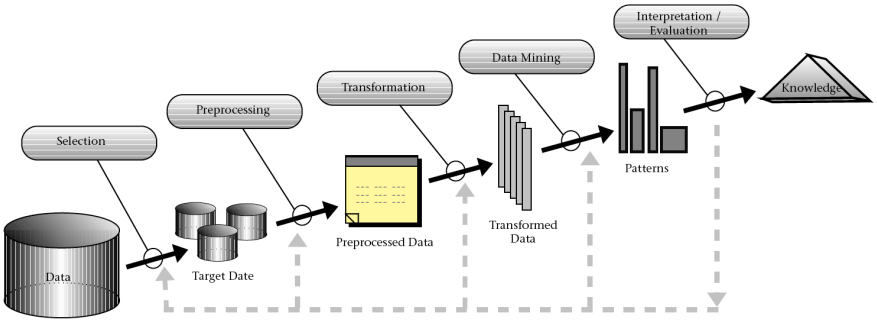


Figura 2.1: Il processo KDD e le sue fasi[4]

3. *Data cleaning* e *preprocessing*, ovvero eseguire operazioni sul *dataset* scelto atte a diminuire il livello di rumore, individuando una strategia per l'identificazione degli *outlier*; in questa fase viene inoltre definita una strategia per gestire i *missing values*.

4. *Data reduction* and *projection*, ovvero trovare gli attributi più significativi a rappresentare i dati, sempre in relazione all'obiettivo finale. In questo step vengono utilizzati algoritmi di *data reduction* per ridurre il numero effettivo delle variabili.

5. Applicazione di metodi di *data mining* quali, classificazione, regressione e clustering per estrarre conoscenza dai dati.

6. Interpretazione della conoscenza estratta, possibilmente con iterazioni successive degli step precedenti; in questa fase è compresa anche la visualizzazione dei pattern/modelli estratti.

7. Consolidazione della conoscenza scoperta.

2.1 Data Selection, Preprocessing and Data Trasformation

Di rado i dati sperimentali sono impiegabili senza elaborazioni in processi di *data mining*; questo fatto è ancora più evidente nel caso di dati *open* come quelli a nostra disposizione. Tra i principali problemi che si riscontrano nei dati vi sono ridondanze, inconsistenze e la presenza di *outlier*. Spesso risulta utile effettuare una prima analisi esplorativa affidandosi a indici statistici, distribuzioni e grafici. Per gli attributi qualitativi e categorici si utilizzano indici quali la moda e strumenti grafici come diagrammi a barre e a torta; mentre per attributi numerici vengono impiegati indici quali media e varianza, e metodi grafici quali istogrammi e boxplot. È bene già in una fase iniziale definire come gestire i *missing values*; questi possono essere risolti con diversi approcci:

- ignorare i valori mancanti;
- eliminare il record;
- stimare e predire il valore attraverso la media o altri metodi statistici.

Il *dataset* inoltre va pulito da elementi duplicati o inconsistenti, la cui probabilità di comparire cresce nel caso di unione di più sorgenti dati. Nella maggior parte dei casi non tutti gli attributi contenuti all'interno di un *dataset* risultano rilevanti ai fini dell'analisi; è così utile ricorrere a tecniche atte a ridurre il numero degli attributi, altresì detta *dimensionalità* del *dataset*, con l'intento di velocizzare le procedure di *data mining* e rendere più efficace la visualizzazione dei risultati. Può inoltre risultare utile generare nuovi attributi in modo da rappresentare meglio le informazioni di maggior interesse, utilizzare aggregazioni o trasformazioni su nuovi spazi. Per poter eseguire con successo alcuni algoritmi di *data mining* può essere necessario discretizzare gli attributi continui, scegliendo il numero di *bin* e gli *split point* più opportuni.

2.1.1 Metodologie per l'outlier detection

In statistica, un *outlier* è un occorrenza tra i dati considerati nell'analisi che risulta distante dalle altre occorrenze[5][6]. Un outlier può essere causato da errori nella trascrizione delle informazioni nella base dati o può identificare un errore sperimentale. In ogni caso è di fondamentale importanza utilizzare metodologie specifiche atte ad identificali in una fase iniziale per escluderli dalle analisi, dato che inquinerebbero certamente i risultati. Nel nostro caso specifico si sono utilizzate tecniche univariate, ovvero che tegono conto di un singolo attributo per volta (gESD, MAD, boxplot) e di tecniche multivariate (DBSCAN).

Tecniche univariate per l'outlier detection

In questa sezione analizzeremo le tecniche univariate per l'*outlier detection*. Nel nostro caso sono state impiegate nella fase preliminare del *preprocessing* come supporto decisionale all'esperto di dominio.
Tra queste citiamo:

- **gESD**: il *generalized Extreme Studentized Deviate*, è un metodo per l'identificazione degli *outlier* introdotto da Rosner nel 1983[7], ed è impiegato per identificare gli *outlier* in un *dataset* dove sussiste l'ipotesi di normalità. Si presenta come un evoluzione del test di Grubbs, test statistico introdotto nel 1950[8] e poi esteso nel 1969[5] e 1972[9] dallo stesso autore. Il test di Grubbs è definito come:

$$G = \frac{max_{\mathrm{i=1,\dots,N}}|Y - \overline{Y}|}{s}$$

  Dove s rappresenta la deviazione standard e $\overline{Y}$ rappresenta la media del campione. Se il massimo G relativo all'$i-esimo$ elemento è più grande del relativo valore tabulato, allora l'elemento è considerato un *outlier*. Il test prosegue fino a che non viene più identificato nessun *outlier* e parte dall'assunzione che

l'osservazione più lontana dalla media dei campioni sia un *outlier*[10]. Il test quindi è in grado di identificare un singolo *outlier* per volta e se usato iterativamente sullo stesso campione, a causa di proprietà degli *outlier* multipli note come *masking* e *swamping*, non tutti gli *outlier* vengono identificati correttamente[11]. Il *masking*, o effetto maschermento, può avvenire infatti quando vengono ricercati troppi pochi *outlier* per uno specifico test; per esempio eseguendo un test per un singolo *outlier* quando nella realtà sono presenti almeno 2 valori da considerare tali, questi *outlier* addizionali potrebbero influenzare il test statistico abbastanza da invalidare il test stesso, così da non permettere l'identificazione di nessun *outlier*. Al contrario lo *swamping* può avvenire quando vengono ricercati attraverso un test statistico troppi *outlier* all'interno di una distribuzione che invece ne contiene pochi, provocando la classificazione di tutti o di nessun punto come *outlier*[12].

Con l'intento di risolvere questi problemi Rosner introduce il gESD, che non necessita di specificare a priori il numero di *outlier* k come nel test di Grubbs; il metodo richiede infatti solo di fissare un upper bound del numero massimo previsto di *outlier* nel dataset in esame. Definendo r l'upper bound, il gESD esegue r separati test: un test per un *outlier*, un test per due *outlier*, e così via fino a r *outlier*[13]. Il test si definisce come:

$$R_{\mathrm{i}} = \frac{max_{\mathrm{i}}|x_{\mathrm{i}} - \overline{x}|}{s}$$

Dove s rappresenta la deviazione standard e $\overline{x}$ rappresenta la media del campione; ad ogni passo si rimuove il campione che massimizza $|x_{\mathrm{i}} - \overline{x}|$ e si ricalcola la statistica con $n-1$ campioni. Il processo viene ripetuto fino a che r campioni sono stati rimossi; si ottengono così r statistiche dai vari test $R_1, R_2, ..., R_r$.

Alle r statistiche corrispondono altrettanti valori critici:

$$\lambda_i = \frac{(n-i)t_{p,n-i-1}}{\sqrt{(n-i-1+t^2_{p,n-i-1}}} \qquad\qquad i = 1,2,...,r$$

Dove t_p, v rappresenta il percentile di una t di Student con v gradi di libertà e

$$p = 1 - \frac{\alpha}{2(n - i + 1)}$$

Il numero di *outlier* viene determinato trovando il più grande i tale per cui $R_i > \lambda_i$. Dalle simulazioni di Rosner è emerso che il numero approssimato di *outlier* trovati è molto accurato a partire da una dimensione del campione di 25 osservazioni. Inoltre il gESD compie degli appropriati aggiustamenti al valore critico basati sul numero degli *outlier* testati.

- **MAD**: il metodo Median Absolute Deviation (MAD) è un metodo di individuazione degli *outlier* molto robusto, riscoperto e reso popolare da Hampel nel 1974[14] che attribuì l'idea a Carl Friedrich Gauss. La mediana (M) è, come la media, una misura di tendenza centrale ma offre il vantaggio di essere poco sensibile alla presenza di *outlier*[15], oltre che non essere per nulla influenzata dalla grandezza del campione preso in esame. Calcolare il MAD inoltre risulta semplice, dato che necessita solo di trovare la mediana della deviazione assolita dalla mediana. Più precisamente il MAD è definito come segue[16]:

  $$MAD = med(|x_i - M)|)$$

  dove

  $$M = med(x_i)$$

  x_i sono le n osservazioni originali e $med(x_i)$ è la mediana delle serie.

- **Boxplot**: introdotto da Tukey nel 1977[17], è una delle più utilizzate tecniche grafiche per visualizzare ed analizzare dataset con un approccio univariato[18]. Se $X_n = \{x_1, x_2, ..., x_n\}$ è il nostro *dataset*, il boxplot si costruisce:

  – disegnando una linea all'altezza del campione corrispondente alla mediana Q_2;

  – disegnando un rettangolo (*box*) che parte dal primo quartile Q_1 fino al terzo quartile Q_3. La lunghezza di questo box risulta così equivalente al range interquartile, che si definisce come $IQR = Q_3 - Q_1$;

  – classificando tutti i punti al di fuori dell'intervallo (*recinto*)

$$[Q_1 - 1.5IQR; Q_3 + 1.5IQR]$$

come potenziali *outlier*, ed evidenziandoli nella grafico;

– disegnando dei *baffi*, ovvero delle linee che partono dai limiti superiore ed inferiore del box fino ai due punti rispettivamente più remoti per lato all'interno del *recinto*.

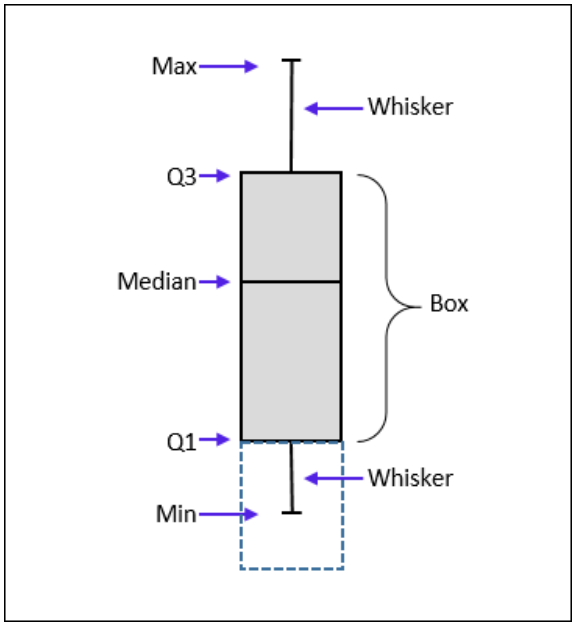


Figura 2.2: Immagine di esempio relativa ad un boxplot.

Tecniche multivariate per l'outlier detection: DBSCAN

In questa sezione analizzeremo la tecnica multivariata utilizzata in questo lavoro per l'*outlier detection*, ovvero il DBSCAN; nel nostro caso è stata utilizzata per pulire gli attributi per cui non sono stati identificati i limiti di validità dall'esperto di dominio, così da poter impiegare efficacemente in seguito algoritmi di clustering.

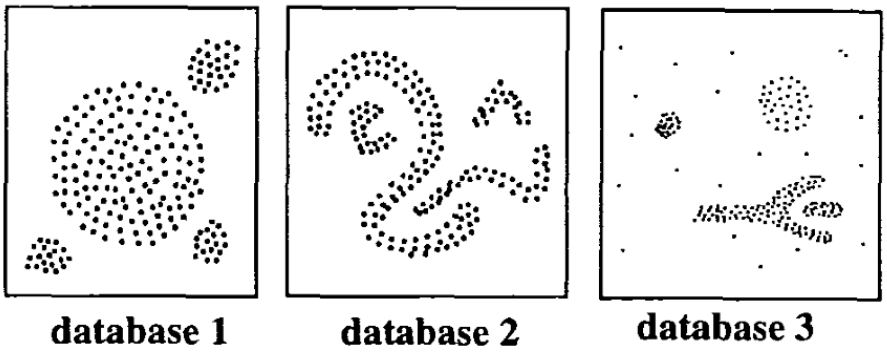


Figura 2.3: La figura mostra 3 esempi di database, per semplicità rappresentati in due dimensioni, dove il DBSCAN ha buone performance[19]

- **DBSCAN** Il DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) è un algoritmo di clustering proposto nel 1996 da Martin Ester, et al.[19]. L'algoritmo è in grado di identificare cluster in grandi *dataset* prendendo in considerazione la densità locale degli elementi del database e necessitando di due parametri di input[20]. Il DBSCAN inoltre può identificare le osservazioni del *dataset* da classificare come rumore tuttavia,come nel caso del nostro *dataset*, cluster che si trovano vicini gli uni agli altri e con densità simile non vengono scissi bene dall'algoritmo. La capacità di individuare gli *outlier* in un *dataset* multivariato rimane comunque una caratteristica fondamentale, che ha fatto si lo impiegassimo anche se limitatamente alla pulizia dei dati.

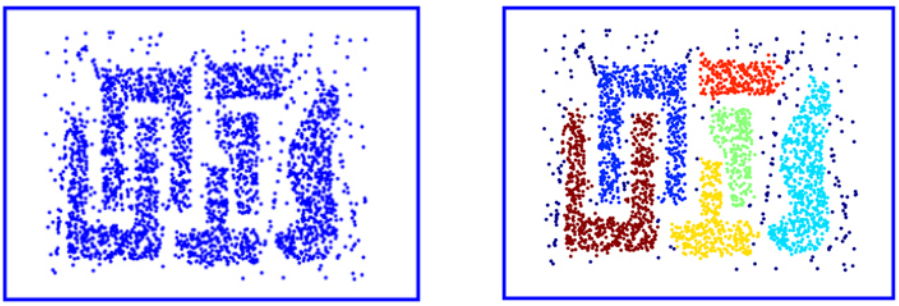


Figura 2.4: La figura mostra il funzionamento dell'algoritmo DBSCAN nel riconoscere cluster non convessi.

Il DBSCAN si basa su sei definizioni e 2 lemmi:

– **Definizione 1:** l'*Eps_neighborhood* di un punto

$$N_{Eps} = \{q \in D | dist(p, q) < Eps\}$$

Un punto per appartenere ad un cluster ha bisogno di avere almeno un altro punto che si posizioni vicino ad esso entro la distanza *Eps*.

– **Definizione 2:** *Directly density-reachable*

Esistono due tipologie di punti che appartengono a un cluster: *border points* e *core points* come si può notare nella figura 2.5.

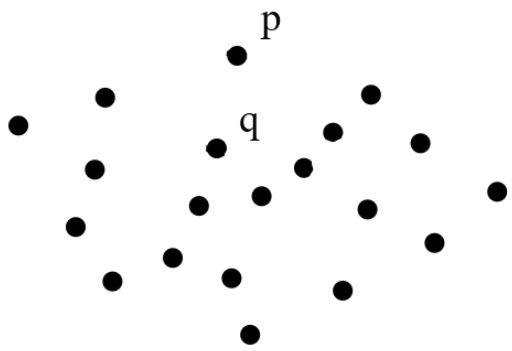


Figura 2.5: Nell'immagine viene evidenziato un esempio di core point (q) e border point (p).

L'*Eps-neighborhood* di un *border point* tende ad avere molti meno punti dell'*Eps-neighborhood* di un *core point*. I *border point* per essere riconosciuti come parte di un cluster devono appartenere all'*Eps-neighborhood* di un *core point* q, come mostrato dalla figura 2.6. Inoltre:

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

Per far si che un punto q sia identificato come un *core point*, questo deve avere un minimo numero di punti all'interno del suo *Eps-neighborhood*.

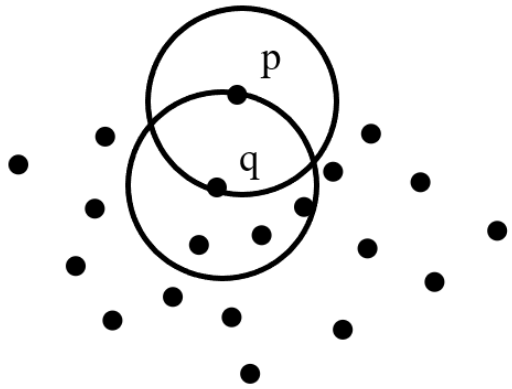


Figura 2.6: In figura p è un punto *Directly density-reachable* da q, ma non è vero il vice versa.

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

- **Definizione 3:** *Density-reachable* Un punto p è *density-reachable* da un punto q, rispettando Eps e $MinPts$ seesiste un a catena di punti $p_1, ..., p_n$, $p_1 = q$, $p_n = p$ tale per cui p_{i+1} è *directly density-reachable* da p_i, come è evidenziato dalla Ffigura 2.7.
- **Definizione 4:** *Density-connected* Un punto p è *density-connected* ad un punto q, rispettando Eps e $MinPts$, se esiste un punto o tale per cui sia p che q sono *density-reachable* da o sempre tenendo in considerazione Eps e $MinPts$.
- **Definizione 5:** *Cluster* Se il punto p fa parte di un cluster C e il punto q è *density-reachable* dal punto p, allora q fa anch'esso parte del cluster C. Inoltre dire che due punti appartengono allo stesso cluster C, è equivalente ad affermare che p è *density-connected* con q.
- **Definizione 6:** *Rumore* Viene identificato come *rumore* il set di punti del database che non appartiene a nessun cluster.

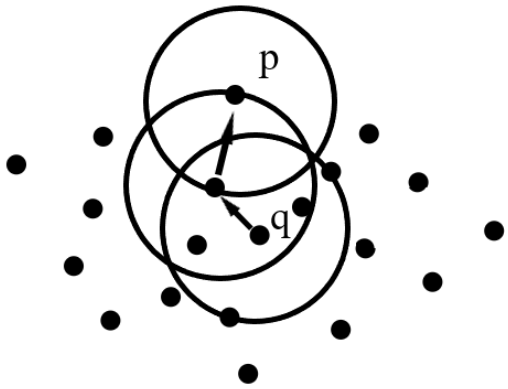


Figura 2.7: In figura p è un punto *density-reachable* da q, ma non è vero il vice versa.

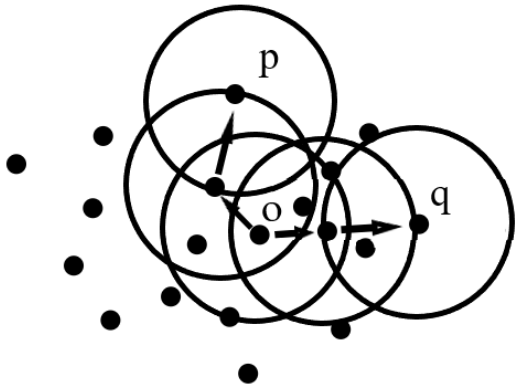


Figura 2.8: Visualizzazione del concetto di *density connectivity*

- **Lemma 1:** Un cluster può essere inizializzato da uno qualsiasi dei suoi *core point* mantenendo la stessa forma.
- **Lemma 2:** Sia p un *core point* di un cluster C. Se il set di punti O *density-reachable* da p, allora C e equivalente al set O.

Per trovare un cluster, DBSCAN inizia con un punto p arbitrario e aggrega tutti i punti *density-reachable* da p. Se p è un *core point*, la procedura crea un cluster. Se p è un *border point* allora nessun punto è *density-reachable* da p e DBSCAN visita il punto successivo del database[19].

2.2 Data Mining

Questa fase del processo KDD consiste nell'applicare algoritmi di *data analysis* che, sotto accettabili limiti ed efficienze computazionali, evidenzino particolari *pattern* all'interno dei dati[4]. Si possono inoltre distinguere due categorie di obiettivi: *Verifica*, dove il sistema si limita a verificare le ipotesi da cui è cominciata l'analisi, e *Scoperta*, dove il sistema trova in modo autonomo *pattern* significativi. Riguardo a quest'ultima categoria, essa viene ulteriormente suddivisa in *predizione*, dove il sistema ricerca pattern con l'intento di predire futuri comportamenti sulla base di un modello, e *descrizione*, dove il sistema ricerca *pattern* con lo scopo di presentarli all'utente in un formato facilmente interpretabile. Molti dei metodi impiegati sono derivati da tecniche di *machine learning* e statistica, come gli algoritmi di classificazione, regressione e clustering. Nelle sezioni successive verranno analizzate le tecniche utilizzate nel presente lavoro.

2.2.1 Algoritmi di Clustering

Sotto il termine *cluster analysis*, o *clustering*, si intendono quei processi ed algoritmi che raggruppano i record di uno specifico *dataset* con lo scopo di individuare *subset* composti di elementi il più possibile omogenei tra di loro; i primi studi in questo campo furono compiuti in antrologia da Driver e Kroeber nel 1932, e successivamente furono introdotti nel campo della psicologia da Zubin nel 1938 e da Robert Tryon nel 1939[21].

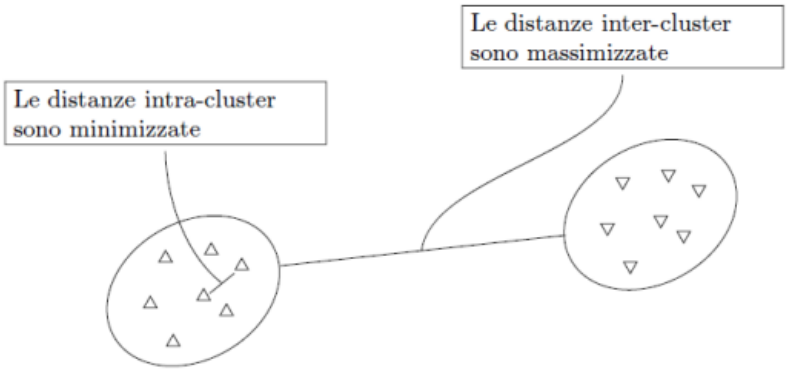


Figura 2.9: Concetti base implementati dagli algoritmi di clustering

Come si può notare dalla figura 2.9 un cluster è tale per cui è presente:

- una grande similarità intra-cluster, ovvero elementi appartenenti ad uno stesso cluster risultano omegenei.
- una bassa similarità inter-cluster, ovvero elementi appartenenti a cluster differenti risultano altamente disomogenei tra di loro

I record vengono discriminati a seconda di una misura di similarità, molto spesso concepita come una misura di distanza in uno spazio multidimensionale. Tra le misura di distanza più comuni si possono citare:

- **Euclidean Distance** la distanza Euclidea tra due osservazioni rappresenta la radice quadrata della somma dei quadrati dei rispettivi valori

$$d = (i, j) = \sqrt{\sum_{k=1}^{n} (X_{ik} - X_{jk})^2}$$

dove $i = (X_{i1}, X_{i2}, ..., X_{in})$ e $j = (X_{j1}, X_{j2}, ..., X_{jn})$ sono due vettori $n-$dimensionali.

- **Squared Euclidean Distance** la distanza Euclidea quadrata si esprime come

$$d = (i, j) = \sum_{k=1}^{n} (X_{ik} - X_{jk})^2$$

La particolarità di questa misura di distanza rispetto alla Euclidea standard è di pesare maggiormente le distanze relative ad osservazioni lontane fra di loro.

- **Manhattan Distance** anche nota come distanza City Block o distanza L_1. La Manhattan Distance è la misura della distanza fra due punti seguendo il percorso tracciato dai due cateti di un ipotetico triangolo in uno spazio bidimensionale. Questa metrica risulta più robusta agli outlier rispetto alle distanze Euclidee spracitate e si esprime come

$$d = (i, j) = \sum_{k=1}^{n} |X_{ik} - X_{jk}|$$

- **Mahalanobis Distance** la distanza di Mahalanobis rappresenta una generalizzazione della distanza Euclidea, dove i pesi delle variabili sono assegnati attraverso la matrice di varianza-covarianza costruita sui campioni, tenendo di conseguenza conto nel calcolo della distanza, della correlazione tra le variabili; questa si calcola come[22]

$$d(a, b) = [(a_i, b_i)^t S^{-1} (a_i - b_i)]$$

dove S^{-1} rappresenta l'inverso della matrice di covarianza.

- **Minkowski Distances** la distanza di Minkowski rappresenta una generalizzazione sia della distanza Euclidea che della Manhattan distance, ed è definita come

$$d = (i, j) = \sum (|X_{ik} - X_{jk}|)^{\frac{1}{q}}$$

dove q rappresenta un numero intero positivo.

- **Cosine Distance** la distanza coseno viene usata spesso nel caso di analisi di dati testuali oltre che essere impiegata come misura di coesione intracluster; questa si calcola come

$$d = \cos(\theta) = \frac{AB}{|A||B|}$$

Gli algoritmi di clustering risultano esaustivi, ovvero sono tali per cui partizionano in classi tutti i record presi in esame, e mutuamente esclusivi, ovvero creano delle partizioni ad intersezione vuota. Per dovere di completezza, esistono anche tecniche denominate di clustering non esclusivo, dove uno stesso elemento può appartenere a più cluster in percentuali variabili, note come soft clustering, altresì detto fuzzy clustering. Un'ulteriore suddivisione delle sopracitate tecniche di clustering prende in esame il tipo di algoritmo utilizzato per dividere lo spazio:

- **Clustering partizionale**: altresì detto clustering non gerarchico o k-clustering, per definire l'appartenneza di un item ad un cluster viene utilizzata una misura di distanza da un punto rappresentativo del cluster stesso (molto spesso il centroide o il medoide), avendo prefissato il numero di gruppi originati dal partizionamento. Si tratta di derivazioni e perfezionamenti del k-means, noto algoritmo di clustering introdotto da MacQueen nel 1967[23].

- **Clustering gerarchico**: viene costruita una gerarchia di partizioni caratterizzate da un numero variabile (crescente, decrescente) di gruppi, visualizzabile mediante una rappresentazione ad albero (dendrogramma), nella quale sono rappresentati i passi di divisione/accorpamento dei gruppi.

- **Clustering basato sulla densità**: come nel noto algoritmo DBSCAN, vengono creati i cluster affidandosi ad una misura di densità spaziale delle osservazioni nello spazio $n-$dimensionale, permettendo, rispetto ad algorimi come il k-means, di riconoscere cluster anche di forma non strettamente convessa.

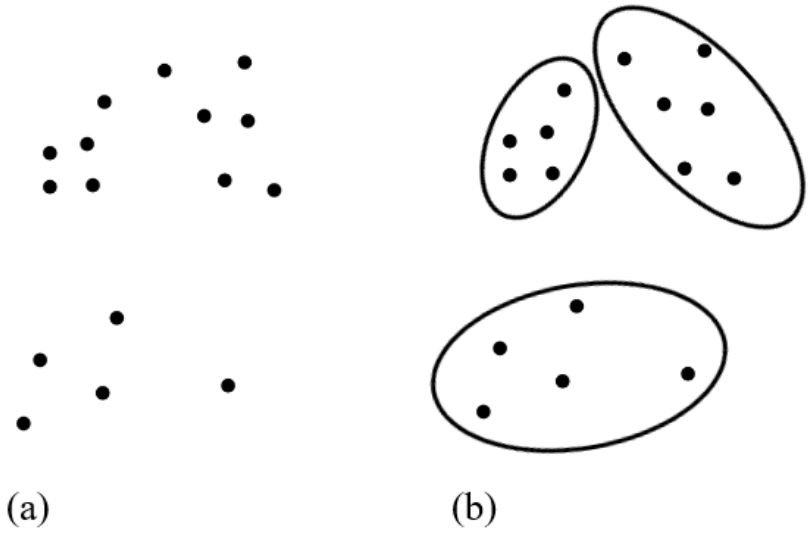


Figura 2.10: La figura mostra il risultato (b) di un algoritmo di clustering partizionale su di un insieme di dati (a)

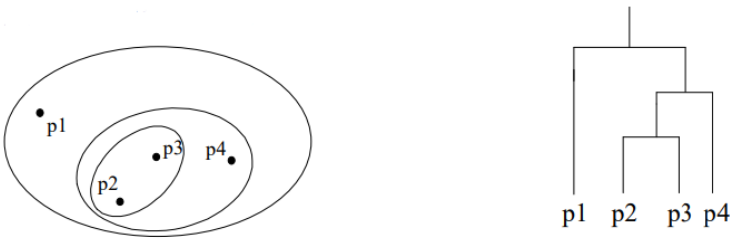


Figura 2.11: Clustering gerarchico e relativo Dendogramma

2.2.2 Alberi di Decisione

Gli Alberi di decisione figurano tra i metodi più usati nel *data mining* per classificare un oggetto in un insieme di classi predefinite, grazie alla loro relativa facilità di implementazione e interpretazione. Questi, si compongono di una struttura ad albero, dove ogni nodo interno rappresenta un sottoinsieme dei record, e i sottoinsiemi finali vengono chiamati nodi foglia. I nodi sono etichettati con il nome della

classe da predire nel caso di un nodo foglia, e con i nomi delle variabili di interesse nel caso di nodi interni. I rami invece, presentano i valori che la variabile *splittata* può assumere. Percorrendo l'albero da un nodo foglia al nodo radice, si possono estrarre le regole che hanno determinato la classe presente nel nodo foglia. Gli alberi di decisione si basano su due concetti fondamentali:

- **Entropia**: viene definita come l'impurità del *dataset*. Se un attributo assume n valori differenti, allora l'entropia di S associata alla $n-$esima classe è definita come:

$$H(S) = -\sum_{i=1}^{n} p_i log_2(p_i)$$

  dove p_i rappresenta la percentuale di S appartenente alla classe i. Un valore di entropia elevata, significa che S possiede una distribuzione uniforme, e solitamente poco interessante; al contrario un'entropia bassa indica che la classe S è composta di pochi valori che portano la maggior parte dell'informazione[24].

- **Information gain**: rappresenta la diminuzione di entropia che uno split su un nodo dell'albero genera, e viene calcolato come:

$$G(S, A) = H(S) - \sum_{v \in Val(A)} \frac{S_v}{S} H(S_v)$$

  dove $Val(A)$ rappresenta il set di tutti i potenziali valori per l'attributo A, e S_v rappresenta il sottoinsieme di S per cui A assume valore v[25]. Questo attributo viene utilizzato per strutturare l'albero di decisione, il quale sceglierà ad ogni iterazione, l'attributo A che massimizza l'*information gain*.

2.3 Interpretazione della Conoscenza Estratta

In questa fase si analizzano i risultati e i pattern estratti dalla fase di *data mining*, tenendo conto degli obiettivi che ci si era prefissati; le metodologie interessate per

la valutazione della bontà dei risultati vanno dal giudizio dell'esperto di dominio all'impiego di tecniche statistiche ed indici. Molto spesso in maniera iterativa si ritorna ai passi precedenti del KDD; questo infatti si compone di svariati cicli di *trial and error*, in cui si testano algoritmi e metodologie differenti, in modo da trovare quelle maggiormente performati sul particolare dominio dei dati su cui si compie l'analisi. In questa fase risulta molto utile affidarsi a tool di visualizzazione della conoscenza; questa molto spesso è resa tramite grafici, e nel caso di dai spaziali come quelli presentati in questo lavoro, tramite mappe per facilitarne l'esplorazione.

Capitolo 3

Framework

In Figura 3.1 è presentata l'architettura di TUCANA (**TU**rin **C**ertificates **ANA**lysis), il *framework* sviluppato per l'estrazione e la visualizzazione di conoscenza utile, a partire dalle certificazioni energetiche relative alla città di Torino.

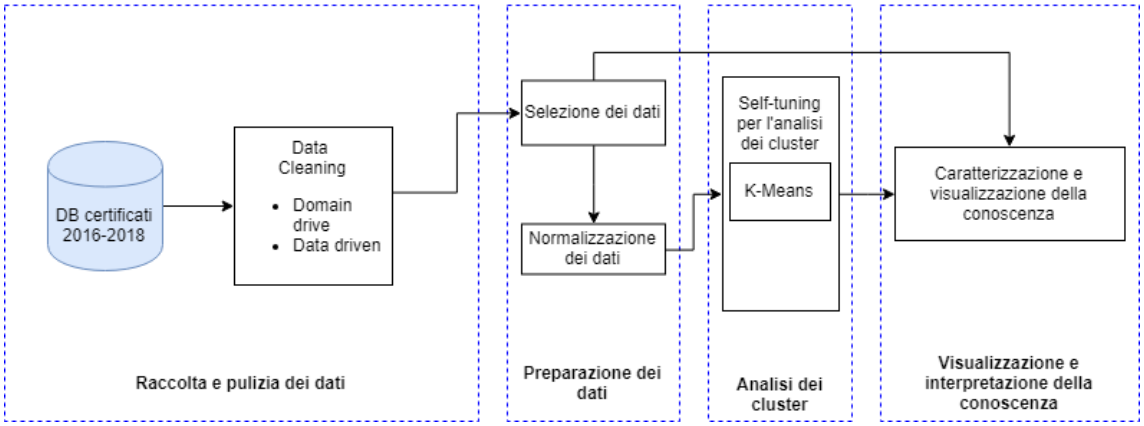


Figura 3.1: Architettura del *framework* TUCANA

Il *framework* è composto da quattro componenti principali:

- **Raccolta ed pulizia dei dati**: in questa fase vengono eseguite procedure di *preprocessing* sui file delle certificazioni energetiche della regione Piemonte forniti dal CSI. Sui dati viene eseguita una prima pulizia sulle variabili fondamentali di valore energetico e fisico con il supporto dell'esperto di dominio. I dati a nostra disposizione appartengono a certificati relativi ad un periodo di tempo che va dal 2016 al primo semestre del 2018.
- **Preparazione dei dati**: in questa fase i dati vengono preparati per essere utilizzati nelle analisi successive. Vengono impiegate tecniche multivariate per l'*outlier detection* e tecniche per la normalizzazione dei valori degli attributi. Questi sono inoltre esplorati attraverso l'uso di mappe.
- **Analisi dei cluster**: in questa fase i certificati vengono raggruppati per caratteristiche fisiche ed energetiche simili, utilizzando l'algoritmo di clustering k-means; viene sviluppato un metodo per agglomerare cluster multipli sulla base di misure di distanza e coesione intracluster per meglio separare gruppi di certificati dalle caratteristiche simili e rilevanti.
- **Visualizzazione della conoscenza**: in questa fase vengono utilizzate le mappe per visualizzare la conoscenza ottenuta dalle fasi precedenti. Gli strumenti di visualizzazione sono stati utilizzati sia durante l'esplorazione dei dati, per migliorare la conoscenza sul dominio, sia in fase di valutazione dei *pattern* estratti attraverso le tecniche di *data mining*.

3.1 Raccolta e Pulizia dei Dati

3.1.1 Raccolta Dati

La prima componente del *framework* è preposta alla raccolta e la pulizia dei dati. in questa tesi , sono stati utilizzati i dati forniti dal CSI-Piemonte, che contengono in

diversi *dataset* in formato CSV le informazioni riguardanti gli attestati di prestazione energetica della regione Piemonte, relativi agli anni dal 2016 al primo semestre del 2018.

Nome file	
dati edifici reali	(a)
dati edifici di riferimento	(b)
dati consumi	(c)
dati energetici	(d)
dati raccomandazioni	(e)

Tabella 3.1: Nella tabella sono illustrati i dataset disponibili; per la nostra analisi sono stati considerati i *dataset* (a), (b) e (d)

Come mostrato nella tabella 3.1, i *dataset* a nostra disposizione trattano informazioni complementari relative ad uno stesso certificato energetico; inoltre, per via edifici ristrutturati o di errori dovuti alla trasmissione degli APE al sistema centrale, sono presenti certificati multipli relativi ad una stessa unità abitativa (identificata con la tripletta *foglio*, *particella* e *subalterno*). Si è scelto così di effettuare la *join* su questi tre attributi conservando, in caso di certificati multipli, quello con data di *upload* più recente. Dopo la *join* , effettuata sui *dataset* (a), (b) e (d) si ottengono circa 110000 certificazioni. Per massimizzare le possibilità di trovare dei *pattern* generalizzabili, si è deciso di concentrare le analisi sulla porzione di *dataset* più densa. Il db è stato così filtrato sulla sola città di Torino, e come destinazione d'uso si sono selezionati solo i certificati appartenenti alla categoria E1(1), ovvero gli edifici adibiti a domicilio a carattere continuativo. All'interno di questo sottoinsieme, sono stati selezionati esclusivamente i certificati riguardanti l'oggetto APE *unità immobiliare*.

3.1.2 Procedura di Pulizia dei Dati

Per estrarre conoscenza utile dai dati a nostra disposizione, i certificati hanno bisogno di essere depurati da errori ed inconsistenze, che comprometterebbero le analisi successive. Questa fase di *preprocessing* sul *dataset* è stata svolta in diverse fasi.

1. **Pulizia degli Indirizzi**

   Il *dataset* a nostra disposizione presenta coordinate geospaziali, indicazione di CAP, indirizzo e numero civico di appartenenza della certificazione energetica in esame; questi attributi, come anticipato nel Capitolo 1, sono di fondamentale importanza per quanto riguarda la visualizzazione dell'informazione tramite mappe, tuttavia i campi spesso risultano errati o incompleti. Prendendo in esame l'attributo Indirizzo inoltre, questo si presenta come un campo testuale libero, caratterizzato da molti errori di battitura e di immissione, che rendono necessaria una *sanitizzazione* avanzata. Per la pulizia dei sopracitati attributi è stato così sviluppato un algoritmo *multistep*, capace di ricostruire e correggere l'informazione, presentato in pseudocodice nell'Algoritmo 1.

   Le procedure di *geocoding*, ovvero l'assegnazione delle coordinate geospaziali dato un indirizzo, sono operazioni che diversi servizi online come Google Maps e OpenStreetMaps forniscono, anche attraverso delle API (*Application Programming Interface*). Questi servizi, nel caso di OpenStreetMaps, sono gratuiti anche se poco affidabili come risultato. Al contrario, nel caso delle Geocoding API [1] di Google, il geocoding risulta molto affidabile ed è capace dato un indirizzo in formato testuale (ad es. *Corso Duca 21*) di ricostruire l'indirizzo completo e di associare ad esso informazioni quali il numero civico più vicino e le coordinate in modo consistente. Nonostante la bontà del servizio, vi è una soglia di richieste REST (*Representational State Transfer*) che possono essere effettuate in un secondo e in una giornata, terminate le quali nel primo caso si viene messi in attesa e nel secondo si paga a consumo. Questi servizi sono pensati infatti per essere incorporati all'interno di applicazioni mobile, non per processare una tantum grandi moli di dati: il tempo per processare l'intero *dataset* sarebbe stato sicuramente inaccettabile, rendendo necessario optare per una soluzione più efficiente.

   Ci si è così appoggiati su di un *dataset* open fornito dal comune di Torino contente il viario della città, completo di vie, numeri civici, CAP e geolocalizzazione[2], che d'ora in poi chiameremo il *viario di Torino* per chiarezza espositiva.

[1] https://developers.google.com/maps/documentation/geocoding/intro

[2] https://www.sciamlab.com/opendatahub/dataset/c_l219_260

Tra le versioni disponibili sul sito è stato scelto il db con coordinate espresse nello standard WGS84 (*World Geodetic System*), dato che è lo stesso utilizzato all'interno dei file delle certificazioni energetiche in nostro possesso. Questo db è stato utilizzato come base verificata ed affidabile con cui confrontare gli indirizzi presenti nelle certificazioni APE; è stato così sviluppato un algoritmo che confrontasse le stringhe nei due db assegnando ad ogni indirizzo nel db dei certificati il suo corrispondete indirizzo più simile presente nel *viario di Torino*. Per la misura di similarità, si è optato per la distanza di Levenshtein normalizzata tra 0 e 1, dove 0 indica totale dissimilarità e 1 totale similarità. A questo punto, è stato possibile settare un valore di similarità soglia sotto il quale, non essendo riuscita con sufficiente grado di confidenza l'associazione dell'indirizzo, si invia una richiesta tramite le Geocoding API. Questa procedura permette di correggere errori nel campo indirizzo, e allo stesso tempo di ricostruire le informazioni mancanti o errate nei campi CAP, numero civico, latitudine e longitudine con un'affidabilità di oltre il 99%.

Distanza di Levenshtein

La distanza di Levenshtein è una metrica utilizzata per misurare la differenza fra due sequenze di caratteri, intesa come il minimo numero di modifiche (inserimenti, cancellazioni e sostituzioni), necessari per trasformare la prima stringa nella seconda; prende il nome dal matematico russo Vladimir Levenshtein, che per primo la propose nel 1965[26]. Nel nostro caso si è utilizzata una versione normalizzata della seguente distanza, presente nella libreria python-Levenshtein[3], che ci ha permesso sia di confrontare agevolmente i risultati provenienti da misure diverse, sia di settare ragionevolmente una soglia di similarità sotto la quale inviare una richiesta di geocoding tramite le Geocoding API. L'indice di similarità fra due stringhe A e B è calcolato come:

$$Levesthtein\_ratio = \frac{Len_{SUM} - L}{Len_{SUM}}$$

[3] https://github.com/ztane/python-Levenshtein

dove L rappresenta la distanza di Levenshtein calcolata come:

$$L = 2NS + NI + NC$$

con

- NS = numero di sostituzioni necessarie per trasformare la stringa A in B
- NI = numero di inserimenti necessari per trasformare la stringa A in B
- NC = numero di cancellazioni necessarie per trasformare la stringa A in B

e dove Len_{SUM} rappresenta la somma delle lunghezze delle due stringhe A e B

$$Len_{SUM} = len(A) + len(B)$$

L'indice *Levesthtein_ratio* sarà tanto più vicino a 1 quanto più le due stringhe A e B saranno simili, e uguale a 1 nel caso di identità.

Di seguito viene mostrato un semplice esempio di calcolo:

$A =$ "ac"

$B =$ "ab"

$L = 2NS + NI + NC = 2 * 1 + 0 + 0 = 2$

$Levesthtein\_ratio = \frac{Len_{SUM} - L}{Len_{SUM}} = \frac{4-2}{4} = 0.5$

Algoritmo di Matching

Come si può notare dall'Algoritmo 1, la procedura inizia con la conversione delle righe estratte dai nostri certificati in caratteri ASCII (*American Standard Code for Information Interchange*); successivamente, attraverso la funzione creoDizionarioDaViario(), viene creata una struttura dati contenente tutte le vie del *viario di Torino*, per ogni via vengono memorizzati i possibili CAP, e per ogni CAP della relativa via i possibili numeri civici con le rispettive

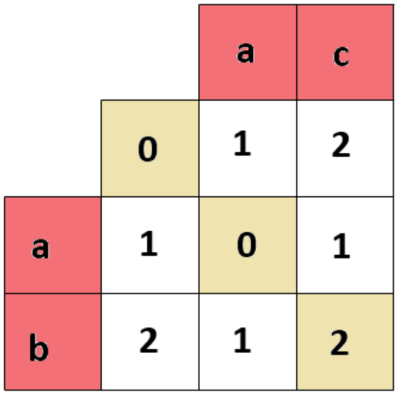


Figura 3.2: Matrice di trasformazione della stringa "*ac*" nella stringa "*ab*"

coordinate, espresse in latitudine e longitudine. Attraverso la funzione creoDizionarioCombinazioniDaViario(), partendo dalle chiavi a livello più alto della precedente struttura dati, ovvero l'elenco delle vie presenti nel *viario di Torino*, si crea una seconda struttura dati contenente come chiave la via e come valore tutte le possibili combinazioni di parole presenti al suo interno; per rendere più chiara la procedura si procederà con un esempio. Prendendo come riferimento la via *"Via Dante Alighieri"* questa sarà per prima cosa spezzata nei *token "Via"*, *"Dante"* e *"Alighieri"*; successimente questi saranno ricombinati in tutte le possibili combinazioni diverse e salvate in una lista [*"Via"*, *"Dante"*, *"Alighieri"*, *"ViaDante"*, *"ViaAlighieri"*, *"DanteVia"*, *"AlighieriDante"*, ecc]. Le nuove combinazioni create vengono salvate senza spazi e saranno le stringhe a essere confrontate con quella presa come input dal campo indirizzo delle certificazioni energetiche.

Preparate le strutture dati di supporto, si procede leggendo da file riga per riga il db delle certificazioni; ad ogni iterazione vengono estratti *via* e *numero civico*, che vengono processati per rimuovere caratteri indesiderati (come segni d'interpunzione); al campo via processato vengono a questo punto rimossi tutti gli spazi presenti, per massimizzare la possibilità di trovare un match con le combinazioni salvate nel $DizViarioCombinazioni$. Iterando sul $DizViarioCombinazioni$ e sulle singole combinazioni di *token* presenti, la distanza di Levenshtein normalizzata viene calcolata per ogni coppia $viaNoSpace \rightarrow token$, memorizzando il massimo valore $maxLev$ della distanza

di Levenshtein trovato .

Se $maxLev$ risulta minore di 0.9 si effettua una query tramite le Geocoding API cercando di recuperare l'informazione completa, altrimenti, tramite la funzione $completaInfo()$ si recuperano numerocivico, CAP, latitudine e longitudine selezionando nel *DizViario* la chiave corrispondente a $viaTrovata$. La procedura ha permesso di risolvere il 97% degli indirizzi tramite il confronto con il *viario di Torino*, e il 2,8% tramite l'uso delle Geocoding API di Google; lo 0,2% degli indirizzi non è stato risolto ed è stato eliminato dal db.

Molto spesso il fatto che gli indirizzi non riuscissero ad essere risolti tramite il confronto con il *viario di Torino*, è stato causato da una non esaustiva registrazione della via. Nel caso infatti che una via come *"Via Dante Alighieri"* fosse memorizzata nel *viario di Torino* come *"Via Dante"*, il confronto con un certificato il cui campo indirizzo fosse *"Via Dante Alighieri"* risulterebbe in un valore di $Levenshtein\_ratio$ molto basso, dovuto alla mancanza della parola *"Alighieri"*, che non sarebbe presente in nessuna della combinazioni presenti per la via *"Via Dante"* nel $DizViarioCombinazioni$. È evidente quindi l'importanza nella buona gestione dei dati open che risulta, anche da questo esempio, di fondamentale importanza.

2. **Scaling**

   Nel *dataset* sono presenti numerose variabili, come ad esempio i rendimenti, espresse con un valore compreso indicativamente tra 0 e 1. Durante un'analisi esplorativa effettuata sui rendimenti di generazione, emissione, regolazione e distribuzione, è stato evidente come molti dei valori erano stati inseriti in forma percentuale al momento della compilazione da parte del certificatore. Su questi attributi, ritenuti fondamentali dall'esperto di dominio, è stato effettuato uno *scaling* di un fattore 100 per i valori oltre il range di validità superiore definito, con l'obiettivo di salvare certificati validi che altrimenti sarebbero andati persi nella fase di eliminazione degli X$outlier$.

3. **Eliminazione degli *outlier***

Algoritmo 1: Pulizia degli indirizzi

input : datasetName, fileViarioName
output: datasetNameCorrect

```
1  conversioneCaratteriASCII(datasetName)
2  DizViario ← creoDizionarioDaViario(fileViarioName)
3  DizViarioCombinazioni ←
     creoDizionarioCombinazioniDaViario(DizViario)
4  for certificato in dataset do
5      AccettaIndirizzo = True
6      via, numCiv, Coordinate ← estraiIndirizzo(certificato, numCiv)
7      if via != vuoto then
8          viaNoSpace, viaProcessata, numCiv ←
             convertiFormatoCorretto(via, numCiv)
9          if viaNoSpace è già stata analizzata then
10             viaTrovata, maxLev ←
                 prendiDaDizVia_Lev(viaNoSpace, numCiv)
11         else
12             viaTrovata, maxLev, CAP, numCiv, coordinate ←
                 cercaViaInDizDaViario(viaProcessata, viaNoSpace)
13     else
14         AccettaIndirizzo = False
15     if AccettaIndirizzo == True then
16         if maxLev < 0.9 then
17             RESULT, IndirizzoTrovato ←
                 queryGeocoding(viaProcessata, NumCiv) if RESULT ==
                 ERROR then
18                 eliminaCertificato(certificato)
19                 AccettaIndirizzo = False
20             else
21                 viaTrovata, CAP, numCiv, coordinate ←
                     infoDaGeocoding(RESULT)
22         viaTrovata, CAP, numCiv, coordinate ←
             completaInfo(viaTrovata, numCiv)
23         resultDb ←
             scriviRigaNuovoDb(certificato, viaTrovata, CAP, numCiv, coordinate)
```

In questa prima fase si è optato per delle tecniche di analisi univariata degli *outlier*, effettuate sugli attributi ritenuti fondamentali dall'esperto di dominio: fattore forma, trasmittanze trasparenti ed opache, rendimenti di generazione, emissione, regolazione e distribuzione. Sono state applicate le metodologie MAD, gESD e boxplot come supporto decisionale all'esperto di dominio, il quale conoscendo le distribuzioni nel nostro caso specifico e i limiti fisici propri di ogni attributo, ha settato dei limiti inferiori e superiori di validità attraverso i quali filtrare il *dataset*.

3.2 Preparazione dei Dati

In questa sezione i dati vengono preparati per gli algoritmi di *data mining*, effettuando diverse procedure di *preprocessing*. Per applicare gli algoritmi di clustering in modo efficace, con l'intento di raggruppare edifici aventi prestazioni termofisiche simili, è necessario innanzitutto effettuare una normalizzazione sui dati. Questa trasformazione sul dominio delle variabili risulta necessaria, dato che gli attributi considerati nell'analisi hanno unità di misure e scale di grandezza differenti. Per far in modo che questa non sia influenzata da outlier e per poter effettuare svariate prove sperimentali, includendo anche gli attributi non ritenuti fondamenti dall'esperto di dominio, si è dovuto ricorrere ad un secondo processo di outlier detection *data driven*, a complemento del precedente approccio *domain driven*. Si è optato per un algoritmo multivariato (DBSCAN)[19], in modo da tener conto delle reciproche influenze fra gli attributi considerati nella specifica analisi.

3.2.1 Normalizzazione dei Dati

Le tecniche di normalizzazione dei dati sono fondamentali per poter applicare in modo efficacie gli algoritmi di *data mining*. Diversi algoritmi infatti (come il k-means) necessitano che le variabili siano confrontabili per poter operare in modo corretto. In

questa sezione si illustreranno brevemente due tecniche di normalizzazione comuni in statistica e nel processamento dei dati.

- **min-max**[27]: la tecnica di normalizzazione min-max fornisce una trasformazione lineare sul dominio originale dei dati ed è espressa come

$$v' = (\frac{v - min_a}{max_a - min_a}) * (newMax_a - newMin_a) + newMin_a$$

  dove min_a e max_a rappresentano rispettivamente il valore minimo e massimo dell'attributo A; la normalizzazione min-max mappa un valore v di A in un valore v' compreso nel range $[newMin_a, newMax_a]$.

- **z-score**[27]: la normalizzazione z-score, definita anche standardizzazione, opera basandosi sulla media e sulla deviazione standard relativa alla distribuzione dell'attributo in esame, e viene definita come

$$v' = \frac{v - \bar{A}}{\sigma_a}$$

  dove $\bar{A}$ rappresenta la media e σ_a rappresenta la deviazione standard dell'attributo A. Si noti come in questo caso non risulti necessario impostare i range minimo e massimo che potrà assumere v'.

3.3 Analisi dei Cluster

In questa sezione i dati preprocessati sono stati analizzati attraverso tecniche di *data mining*, in particolare si è applicato l'algoritmo K-Means per clusterizzare il *dataset*, ovvero dividerlo in gruppi omogeni secondo le caratteristiche termofisiche fondamentali degli edifici individuate dall'esperto di dominio.

3.3.1 K-Means

Il termine *K-Means* fu coniato per la prima volta nel 1967 da James MacQeen[23], anche se l'idea risale al 1957 ed è attribuita a Hugo Steinhaus[28]. Il primo algoritmo standard fu proposto da Stuart Llyod nel 1957, sebbene non fu pubblicato prima del 1982[29]. L'algoritmo, come illustrato nella sezione 2.2.1, fa parte della famiglia delle tecniche partizionali non supervisionate, ed è una delle tecniche più largamente utilizzate grazie anche alla sua semplicità[30]. Si basa sul concetto di centroide, ovvero il centro di massa di un cluster calcolato come media del vettore n-dimensionale degli attributi considerati nel clustering; i centroidi vengono generati random alla prima iterazione e determinano la definizione dei cluster, creati assegnando ogni punto del *dataset* al centroide più vicino secondo una metrica di distanza (vedi 2.2.1 per una definizione delle varie metriche); i centroidi a questo punto vengono ricalcolati con riferimento ai nuovi cluster appena formati, iterando il processo fino a che i centroidi non cambiano entro una certa soglia massima prestabilita.L'algoritmo è classificato come NP-Hard (*Non-deterministic Polynomial-time Hardness*) e la sua complessità si esprime come

$$O(n * K * I * d)$$

dove n identifica il numero di punti nel *dataset*, K il numero di cluster specificato come output, I il numero di iterazioni e d il numero di attributi su cui è compiuta l'analisi.

L'algoritmo ha come obiettivo la minimizzazione della seguente funzione obiettivo

$$W(S, C) = \sum_{k=1}^{k} \sum_{i \in S_k} \|y_i - c_k\|^2$$

dove S è una partizione del *dataset* originata dal k-Means rappresentata dai vettori

$y_i (i \in I)$ all'interno dello spazio n-dimensionale delgli attributi selezionati per l'analisi, relativo a cluster S_k non vuoti, aventi intersezione vuota, ognuno dei quali con un centroide identificato con c_k dove $k = 1, 2, ..., K$[**FW:kMeans2013**]. Il numero k di cluster in cui dividere il *dataset* va scelto preventivamente e fornito come input all'algoritmo; esistono diverse euristiche in grado di fornire un'indicazione sul numero di k più rappresentativo, come per esempio

- **Rule of thumb** è un euristica molto semplice ed applicabile a ogni tipologia di *dataset* che consiste nel determinare k come

$$k \approx \sqrt{\frac{n}{2}}$$

  dove n rappresenta il numero di dati all'interno del *dataset* considerato.

- **Elbow Method** rappresenta il metodo che per primo fu utilizzato in letteratura per la determinazione di k e viene chiamato, per via della sua visualizzazione grafica, il metodo del gomito[31]. Consiste nel calcolare l'SSE (*Sum of Squared Error*) definito come

$$SSE = \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - m_i\|^2$$

  dove x_j rappresenta un osservazione nel cluster C_i e m_i rappresenta il centroide relativo a C_i, partendo da $k = 2$ ed incrementando k di un'unità per ogni iterazione, per poi plottare su di un grafico i valori assunti dall'SSE al variare di k, come mostrato in Figura 3.3. Solitamente, mano a mano che il k sale diminuisce il valore dell'SSE, di conseguenza il miglior k viene selezionato graficamente come il valore per il quale la curva presenta la più alta diminuzione dell'SSE prima di stabilizzarsi[32][33] (in Figura 3.3 il valore identificato per k è $k = 3$). Il metodo essendo un euristica non identifica il valore di k perfetto, ma al contrario evidenzia un punto di partenza attorno al quale iniziare ad eseguire gli esperimenti, variando il valore di k nell'intorno di valori identificato dal *gomito*.

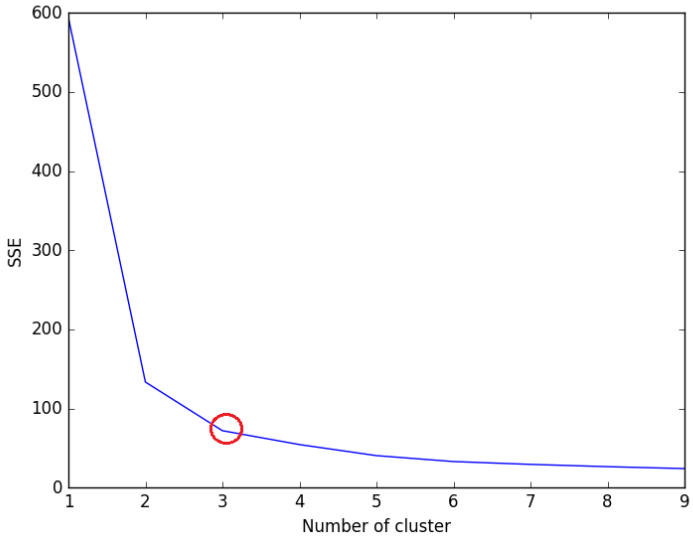


Figura 3.3: Grafico che mostra il variare dell'SSE al variare del numero di cluster k.

Criticità del K-Means

La scelta iniziale dei centroidi può determinare risultati diversi in diversi run di K-Means; si ricorda infatti che i centroidi iniziali sono scelti in modo randomico dall'algoritmo. Una soluzione a questo problema è eseguire il k-Means più volte, mediando i risultati ottenuti nelle diverse prove, inficiando però di contro con il tempo di esecuzione su grandi *dataset*. Inoltre, come si può notare dalla Figura 3.4, il k-Means crea esclusivamente cluster di forma convessa, non risultando di conseguenza adatto come altri metodi (ad esempio DBSCAN e Gaussian Mixture Modeling) per identificare cluster dalle forme complesse, o dove i cluster differiscono molto per dimensione o densità. Una possibile soluzione a quest'ultimo problema ci viene suggerita dalla Figura 3.5. Si può notare visivamente come sia molto chiaro quali cluster aggregare per ricostruire il giusto partizionamento dei gruppi presenti nel *dataset*; per raggiungere lo stesso risultato programmaticamente nel lavoro qui presentato si è optato per un' approccio in due step:

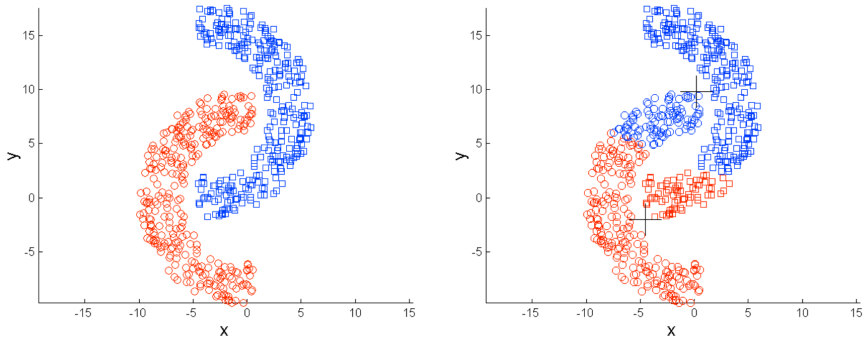


Figura 3.4: Limiti del K-Means nel riconoscere cluster non convessi[34]

- **Euclidean Distance** per prima cosa si è generato un clustering sufficientemente fitto scegliendo opportunamente grande il valore di k. Successivamente si sono calcolate le distanze Euclidee fra i centroidi finali del k-Means, aggregando di volta in volta i due cluster più vicini.

- **Silhouette index**[35] per affinare la strategia di aggregazione si è inoltre calcolato ad ogni iterazione l'indice di Silhouette come misura di coesione intracluster, in modo da evitare di unire due cluster con centroidi *vicini*, ma con coesioni intracluster molto diverse. Si è settato un parametro $maxDiffSil$ che esprime la massima differenza di Silhouette ammissibile affinché due cluster identificati come i più vicini all'n-esima iterazione si uniscano.

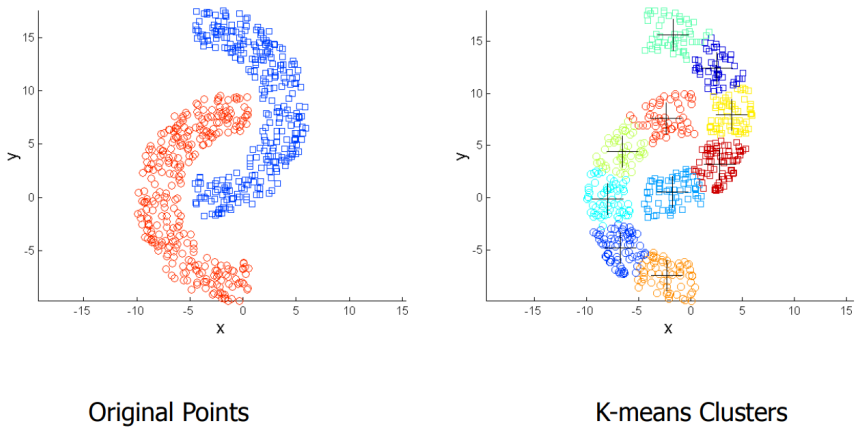


Figura 3.5: Riaggregazione di cluster originati da un k elevato[34]

3.4 Visualizzazione ed Interpretazione della Conoscenza Estratta

Lo scopo di questa ultima componente dell'architettura sviluppata, è stato quello di visualizzare l'informazione estratta dall'esplorazione del *dataset* e dalle tecniche di *data mining*, attraverso strumenti facilmente interpretabili anche dai non addetti ai lavori. Questo è stato possibile attraverso la creazione di mappe interattive e navigabili, che uniscono alla completezza dell'informazione fornita, un livello di fruibilità sicuramente maggiore rispetto a grafici e diagrammi tipici della statistica che, per la loro natura altamente settoriale, risultano di più difficile interpretazione. I dati a nostra disposizione, essendo geolocalizzati, si prestano molto bene a questo tipo di rappresentazione permettendo l'utilizzo di diverse tipologie di mappe; queste sono state impiegate insieme, garantendo in un'unica soluzione diversi livelli di dettaglio a seconda del grado di zoom selezionato dall'utente.

- **Mappe Coropletiche**

  La prima tipologia di mappe è rappresentata dalle mappe *coropletiche*; queste furono in origine proposte da Charles Dupin[36], e identificano delle mappe

tematiche dove le aree delimitate vengono colorate secondo schemi di colore relativi a calcoli su dati statistici aggregati su di un determinato attributo. La Figura 3.6 mostra un esempio di mappa coropletica relativo alla città di New York[37]; in questo particolare contesto le aree sono relative ai vari zip code della città e colorate secondo l'intensità di consumo elettrico durante il 2012. Si può notare come sia immediato identificare l'area di Times Square come quella più energivora, rispetto ad esempio ad aree più periferiche come il Queens. Nel nostro caso si sono usate come delimitazioni delle aree due livelli di dettaglio: le circoscrizioni di Torino e gli isolati.

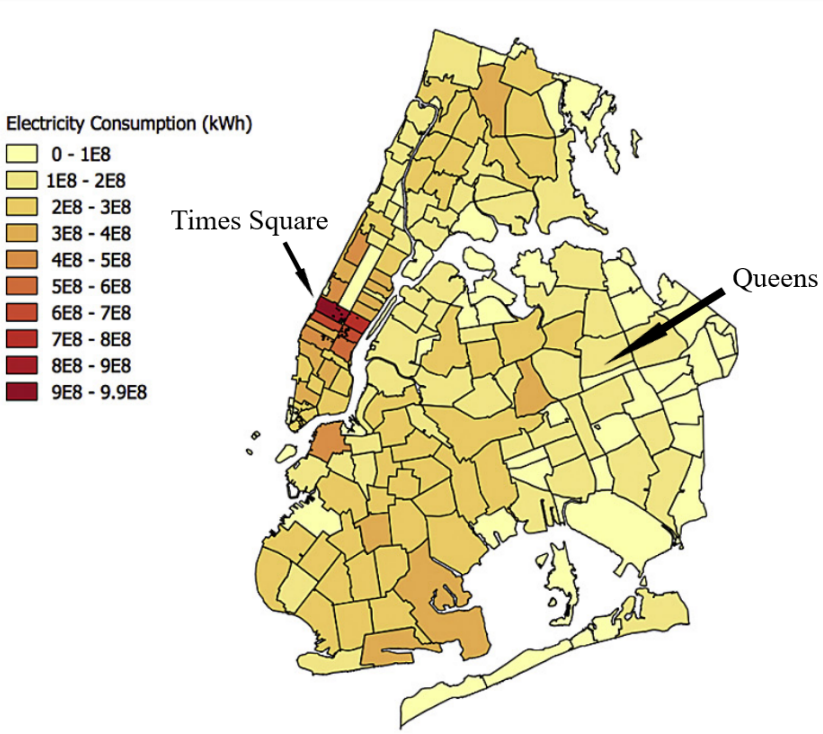


Figura 3.6: Mappa coropletica che mostra il consumo di energia elettrica relativo alla città di New York[37]

- **Mappe Scatter**

  Mentre le mappe coropletiche sono servite per poter mostrare dell'informazione aggregata, si sono utilizzate delle mappe di tipo scatter per poter visualizzare

il dettaglio più fine presente nella nostra base dati, ovvero il singolo certificato; questi sono stati identificati in modo preciso sulla mappa grazie alle coordinate geospaziali, e rappresentati per via di marker, come mostrato in Figura 3.7.

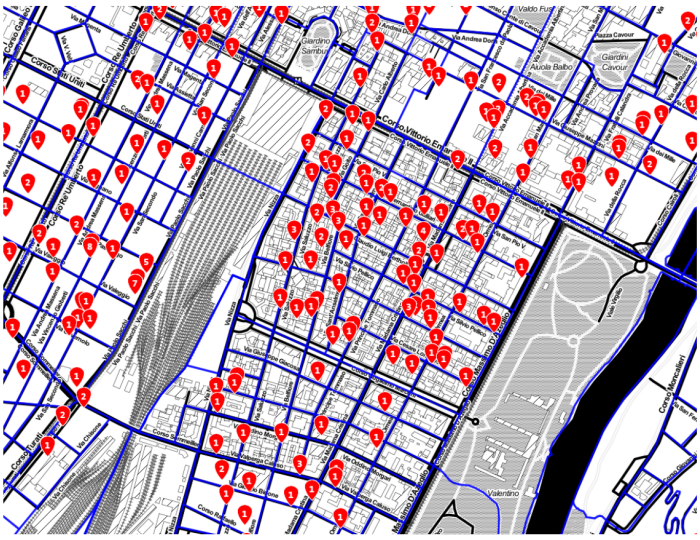


Figura 3.7: Mappa scatter dove ogni marker rappresenta uno o un insieme di certificati.

3.4.1 Realizzazione delle Mappe

Per la realizzazione delle mappe presentate in questo lavoro si è scelto di utilizzare la libreria Python *Folium*[4]. Questa funziona da wrapper per la libreria JavaScript *Leaflet*[5], responsabile della creazione delle mappe dinamiche in formato html (HyperText Markup Language). Le analisi e le elaborazioni dei dati vengono effettuate interamente in Python, e successivamente *Folium* si occupa di generare automaticamente del codice html e JavaScript per la visualizzazione delle mappe dinamiche.

[4] https://pypi.org/project/folium/

[5] https://leafletjs.com/

Il primo passaggio necessario per poter *disegnare* le mappe coropletiche, è stato quello di passare a *Folium* attraverso il metodo *folium.choropleth()* l'attributo di interesse e le aree su cui calcolare le statistiche da rappresentare sulla mappa; queste aree sono state fornite in input attraverso un file GeoJSON, ovvero un file JSON (*JavaScript Object Notation*) con una struttura specifica, in grado di memorizzare dati geospaziali, e contenente punto per punto il disegno geolocalizzato di ogni area poligonale. Sono stati utilizzati due tipi di GeoJSON a diversi livelli di dettaglio, il primo contenete come aree le circoscrizioni[6] e il secondo contenente gli isolati di Torino[7]. Per il GeoJSON relativo agli isolati, non essendo questi indicati nella nostra base dati, è stato necessario sviluppare un algoritmo che andasse a mappare ogni certificato con uno specifico isolato, come mostrato nell'Algoritmo 2.

Algoritmo 2: Matching degli isolati con i certificati

```
input : isolatiTorino, dbCertificati
isolatiTorino ← associaIdUnivoco(isolatiTorino)
poliIsolati ← trasformaInPoligoni(isolatiTorino)
for certificato in dbCertificati do
    for poligono in poliIsolati do
        if èNelPoligono(certificato) == True then
            dbCertificati ← scriviIdPoligono(poligono)
            break
```

Ad ogni isolato presente nel file GeoJSON è stato innanzitutto assegnato un identificativo univoco, per poi essere trasformato in area poligonale attraverso la libreria Python *Shapely*[8] [9]. A questo punto si è andato a verificare per ogni certificato presente nel nostro *dataset*, e scorrendo i poligoni appena creati, se il certificato si trovasse o meno all'interno di uno specifico poligono.Il secondo passaggio utile a disegnare delle mappe coropletiche coerenti con i dati in ingresso, è stato quello di modificare il metodo di *Folium folium.choropleth()*; questo infatti nativamente colora una zona tenendo conto solo dell'ultimo dato in input appartenente a quella determinata area.

[6] https://telemaco87.carto.com/tables/circoscrizioni_geo/public

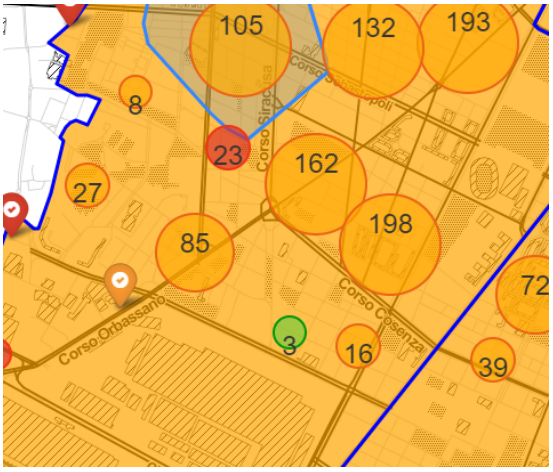
[7] https://andria.carto.com/tables/torino/public/map

[8] https://pypi.org/project/Shapely/

[9] https://www.lfd.uci.edu/ gohlke/pythonlibs/#shapely

Si è così proceduto implementando il calcolo della media su ogni area, che questa si tratti di una circoscrizione o di un isolato, colorando di grigio scuro le zone senza certificati e di grigio chiaro le zone contenti un numero di certificati inferiore a 4, come mostrato nella Figura 3.8(a). Procedendo con l'implementazione, si è potuto notare come la mappa scatter, nonostante sia corretta da un punto di vista formale, presenti non pochi lati negativi dal punto di vista della visualizzazione; mostrare infatti sulla mappa una grossa mole di dati come nel nostro caso, si traduce in una nuvola di marker indistinti, che oltre a non essere facilmente leggibili se non al livelli di zoom elevati, appesantiscono anche il file, non permettendo una agevole navigazione interattiva. Si è così pensato di introdurre nella visualizzazione un nuovo tipo di marker attraverso un plugin di *Leaflet* denominato *Marker Cluster*; come suggerisce il nome, questo nuovo marker aggrega i singoli marker vicini entro un certo raggio espresso in pixel, permettendo di rendere la mappa più leggibile e al tempo stesso informativa; aumentando il livello di zoom i *Marker Cluster* si dividono e diminuendolo si aggregano. È stata così implementata la media pesata sulla cardinalità di certificati contenuti all'interno di ogni singolo *Marker Cluster*, che è stato anche reso più o meno grande a seconda della cardinalità di certificati al suo interno (Figura 3.8(b)); questo ha permesso di dare un'indicazione di magnitudine relativa all'importanza del valore medio mostrato.

(a) Rappresentazione degli isolati e relativa coropletica



(b) *Marker Cluster* di diversa grandezza e valore medio

(c) Zoom massimo su di un *Marker Cluster* rappresentante diversi certificati appartenenti ad unità immobiliari di uno steso edifico

Figura 3.8: Caratteristiche delle mappe realizzate.

Capitolo 4

Risultati Sperimentali

In questo capitolo verranno analizzati i risultati sperimentali ottenuti nelle varie fasi dell'architettura; per prima cosa verranno analizzati gli esiti delle operazioni di selezione degli attributi e di pulizia degli indirizzi, mostrando esempi del funzionamento dell'algoritmo presentato in 3.1.2. Successivamente si presenteranno i risultati dell'*outlier detection domain driven* operata su indicazione dell'esperto di dominio. Nella fase successiva si mostreranno i risultati del clustering, oltre che un esempio di visualizzazione della conoscenza estratta dalle tecniche di *data mining* attraverso le mappe.

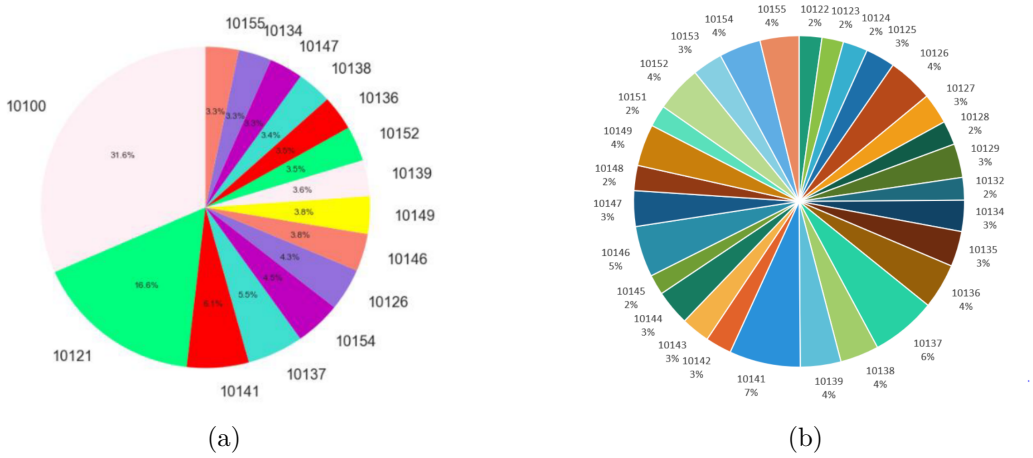
4.1 Raccolta ed integrazione dei dati

Come operazione preliminare si è effettuata un'analisi esplorativa del *dataset* dei certificati. Si è potuto constatare come alcuni attributi presentassero un'elevata percentuale di campi non valorizzati, che sono stati rimossi nel caso rappresentassero oltre il 70% dell'informazione per lo specifico attributo preso in esame. Al termine

dell'operazione è stato così possibile ottenere un *dataset* di 70 attributi. Successivamente si è deciso di concentrarsi sulla sezione più densa di db, che costituisce circa il 90% dei certificati, filtrando il *dataset* per la sola città di Torino, per oggetto APE *unità immobiliare* e destinazione d'uso E1(1). Si sono così ottenuti circa 30000 certificati su cui condurre le analisi successive.

4.1.1 Algoritmo di pulizia degli indirizzi

Si è iniziato anche in questa fase con un'analisi esplorativa dell'informazione presente nel db, notando, come si evince dalla Figura 4.1(a), che la maggior parte dei CAP registrati nelle certificazioni in nostro possesso apparteneva alla classe 10100, CAP generico per la città di Torino e non più in vigore.



(a) (b)

Figura 4.1: Distribuzione dei CAP prima (a) e dopo (b) l'applicazione dell'Algoritmo di pulizia degli indirizzi.

Per questi circa 9000 *record* non si conosceva con precisione il CAP di appartenenza e si è notato che l'informazione relativa alla geolocalizzazione contenuta nei campi Latitudine e Longitudine spesso non risultava affidabile. Un'ulteriore analisi sul campo Indirizzo inoltre, ha fatto emergere le criticità legate al suddetto campo codificato come testuale libero, il quale conteneva informazioni in formato non

standardizzato e spesso errate, come ad esempio errori di battitura. L'applicazione quindi dell'Algoritmo descritto in 3.1.2 , ha permesso di correggere le informazioni errate nei campi Indirizzo, Numero Civico, CAP, Latitudine e Longitudine, oltre che di completare le informazioni mancanti per circa il 99,8% dei certificati. Come si può notare dalla Figura 4.1(b), la situazione relativa alla distribuzione dei CAP di Torino all'interno del *dataset* dopo la pulizia è sensibilmente migliorata, evidenziando una distribuzione pressoché omogenea all'interno del db.

Nella tabella 4.1 sono mostrati alcuni esempi riguardanti l'algoritmo di associazione tra le vie fornite nel campo indirizzo dei certificati presenti nel nostro *dataset*, indicato come *Via Originale*, e le vie presenti nel *viario di Torino* attraverso la distanza di Levenshtein. Vengono mostrate e commentate per prime quelle vie per cui il processo è andato a buon fine, ovvero dove il valore indicato come *Lev_ratio* risulta maggiore di 0.9.

Via Originale	**Via Processata**	**Numero Civico**	**Numero Civico Result**	**Result**	**Lev_ratio**
VIA SANTHI??	VIASANTHI	20	20	VIA SANTHIA'	0.9
VIA DAUBREE	VIADAUBREE	4	4	VIA ADOLPHE DAUBRE'E	0.952380952
VIA CAVOUR (BENSO DI)CAMILLO	VIACAVOURBENSODICAMILLO	34	34	VIA CAMILLO BENSO CONTE DI CAVOUR	0.954545455
VIA GUIDO RENI N. 96/31 H	VIAGUIDORENIN		96	VIA GUIDO RENI	0.96
VIA OXILIA NINO N?? 15	VIAOXILIANINON		15	VIA NINO OXILIA	0.962962963
GABETTI 12	GABETTI		12	CORSO GIUSEPPE GABETTI	1
VIA SANT'ANSELMO 40, TORINO	VIASANTANSELMO		40	VIA S. ANSELMO	1

Tabella 4.1: Esempi di matching con il *viario di Torino*

Il campo *Via Processata* identifica la stringa che viene passata alla funzione di matching spiegata nell'Algoritmo 1; come si può notare vengono eliminati tutti i segni di interpunzione, gli spazi e i caratteri speciali. Il processo è in grado di correggere errori nella codifica degli accenti (per es. *via santhi??* in *via santhià*) e di identificare correttamente i numeri civici inseriti erroneamente all'interno del campo Indirizzo (ad es. *via oxilia nino n.?? 15*), ricostruendo l'informazione mancante. Indirizzi come *Via Sant'Anselmo* inoltre sono risultati complicati da riconoscere dal sistema per via del fatto che nel *viario di Torino* questi sono memorizzati come *Via S. Anselmo*. La soluzione a questo problema è stata quella di aggiungere nel dizionario con tutte le occorrenze delle varie combinazioni di parole per una determinata via, anche

tutti i possibili modi in cui può essere scritta una via contenente un *santo* nel nome (*san,sant,santa,santo*). La stessa cosa è stata fatta anche per parole quali *piazza, piazzale, strada, corso* e i numeri romani.

Nella Tabella 4.2 sono invece presentati degli esempi per quanto riguarda la risoluzioni di indirizzi attraverso le Geocoding API di Google.

Via Originale	**Result Levenshtein**	**Lev_ratio**	**Result Geocoding**
VIA EANDI	VIA CANDIA	0.823	VIA ABATE VASSALLI EANDI
VIA RAFFAELLO MORGHEN	VIA RAFFAELLO LAMBRUSCHINI	0.774	VIA RAFFAELLO MORGHEN
VIA ASTI - EDIFICIO F	VIA CASTELDELFINO	0.678	VIA ASTI
VIA CAROSSIO	VIA CARISIO	0.857	STR DEL CAROSSIO
STRADA PROVINCIALE DI LANZO	STRADA S. VINCENZO	0.736	VIA LANZO
TORINO - VIA ASTI	VIA TRINO	0.761	VIA ASTI
VIA GAMBA	VIA GAMBASCA	0.842	CORSO ENRICO GAMBA
ROVIGOVIA ROVIGO	VIA ROVIGO	0.750	VIA ROVIGO

Tabella 4.2: Esempi di risoluzione di indirizzi attraverso Geocoding API

Come si può notare per i valori di LEV_ratio inferiori a 0.9 la via $ResultLevenshtein$, originata dall'algoritmo di matching, non viene considerata attendibile, procedendo con una richiesta attraverso il servizio online. Si può notare come il processo riesca a riconoscere vie che non sono state identificate perché non presenti nel *viario di Torino* (ad es. *Via Raffaello Morghen*), o che mettono in crisi la metrica di assegnazione del valore *Lev_ratio*, come ad esempio *Strada Provinciale di Lanzo*; quest'ultimo esempio fa notare come il valore di *Lev_ratio* sia influenzato dalla lunghezza delle parole (*strada* vs. *via*), non riuscendo a riconoscere correttamente l'indirizzo, comunque risolto dalle Geocoding API.

4.1.2 Scaling dei Dati ed Eliminazione Outlier

Attraverso l'esperto di dominio si sono così identificati gli attributi principali su cui filtrare il *dataset*, ritenuti fondamentali per considerare termofisicamente valido il certificato (vedi 1.2). Questi sono:

- Fattore forma $[m^{-1}]$

- Trasmittanza Trasparente $[\frac{W}{m^2K}]$
- Trasmittanza Opaca $[\frac{W}{m^2K}]$
- Rendimento di Generazione
- Rendimento di Regolazione
- Rendimento di Emissione
- Rendimento di Distribuzione

Il 30% circa dei certificati tuttavia presentava nei rendimenti valori non compresi fra 0 e 1, ma espressi in percentuale. Si è così proceduto con lo scalare i valori dei rendimenti di un fattore 100 per tutti quei valori maggiori del limite superiore di validità identificato dall'esperto di dominio. Lo scaling effettuato risulta ininfluente ai fini del successivo filtro applicato sul *dataset*, dato che un valore fuori dai range di validità mantiene questa caratteristica anche dopo la trasformazione. Solo nel caso questo valore esprima effettivamente una percentuale, una volta scalato ricadrà all'interno dei range.

Si sono successivamente applicate, a supporto dell'esperto di dominio, le tecniche di *outlier detection* univariate MAD, gESD e boxplot, individuando la percentuale di *outlier* e i range di validità identificati dai rispettivi metodi.

METODO	FATTORE FORMA	TRASMITTANZA OPACA	TRASMITTANZA TRASPARENTE	RENDIMENTO DISTRIBUZIONE	RENDIMENTO EMISSIONE	RENDIMENTO GENERAZIONE	RENDIMENTO REGOLAZIONE
GESD	0,29 %	0,14 %	0,01 %	1,62 %	2,43 %	1,63 %	1,75 %
MAD	2,69 %	1,23 %	0,02 %	42,45 %	41,91 %	14,79 %	41,73 %
BOXPLOT	2,06 %	3,50 %	0,05 %	1,62 %	2,43 %	14,92 %	1,75 %

Figura 4.2: Percentuale di *outlier* identificati da MAD, gESD e boxplot

Come mostrato in Figura 4.2, si può notare che il metodo gESD sia quello che identifica il minor numero di *outlier*, mentre il MAD sia quello che identifica la percentuale maggiore. Prendendo in considerazione anche la Figura 4.3, si nota che,

per quanto riguarda i rendimenti, i tre metodi si comportano in maniera differente; nello specifico il MAD è stato il metodo che meglio di tutti è riuscito ad individuare degli *outlier*, dato che questi attributi possono assumere un valore che solitamente varia tra 0 e 1 in via teorica, e che all'atto pratico variano tra un minimo di 0.5 ad un massimo di 1.1. Rivolgendo invece l'attenzione a fattore forme e trasmittanze si può notare come tutti e tre i metodi riescano ad individuare range di validità credibili.

	GESD				MAD				BOXPLOT			
	Range di esclusione		Range di validità		Range di esclusione		Range di validità		Range di esclusione		Range di validità	
	MIN	*MAX*	*MIN*	*MAX*	*MIN*	*MAX*	*MIN*	*MAX*	*MIN*	*MAX*	*MIN*	*MAX*
FATTORE FORMA	1,50	42,96	0,00	1,47	0,97	42,96	0,00	0,97	1,03	42,96	0,00	1,03
TRASMITTANZA OPACA	3,05	5,65	0,00	3,03	0,00	5,65	0,02	2,35	0,00	5,65	0,29	2,08
TRASMITTANZA TRASPARENTE	25,01	25000,00	0,00	9,17	9,17	25000,00	0,00	8,98	7,97	25000,00	0,00	7,58
RENDIMENTO DISTRIBUZIONE	81,00	100,00	0,00	1,21	0,00	100,00	0,56	1,21	81,00	100,00	0,00	1,21
RENDIMENTO EMISSIONE	9,60	131,83	0,00	1,10	0,00	131,83	0,63	1,10	9,60	131,83	0,00	1,10
RENDIMENTO GENERAZIONE	11,42	299,89	0,00	8,78	1,78	299,89	0,00	1,76	1,64	299,89	0,00	1,62
RENDIMENTO REGOLAZIONE	3,43	100,00	0,00	2,16	0,00	100,00	0,57	1,12	3,43	100,00	0,00	2,16

Figura 4.3: Range di validità identificati da MAD, gESD e boxplot

Attraverso questa analisi preliminare e seguendo un approccio *domain driven*, i range di validità sono stati fissati dall'esperto di dominio filtrando il database in due step: per prima cosa si sono effettuati i filtri su fattore forma e trasmittanze e successivamente si sono applicati i filtri relativi ai rendimenti. Sono state inoltre effettuate delle analisi sui certificati eliminati ad ogni passo della procedura, per indagare le ragioni della cancellazione del record ed identificare eventuali soluzioni. Applicando il primo filtro sui range di validità di trasmittanze e fattore forma, si sono eliminati circa il 7.6% dei certificati; questi sono stati cancellati anche quando solo uno dei tre attributi fosse risultato fuori dai range di validità. L'analisi e il confronto del *dataset* prima e dopo l'applicazione del filtro, ha potuto evidenziare come circa il 99.6% dei certificati presentava le trasmittanze entro i range di validità (Figura 4.4), facendo così emergere come l'eliminazione delle occorrenze fosse dovuta a valori di fattori forma fuori dai limiti di validità.

Successivamente si sono applicati i filtri sui rendimenti, campi che presentano una

Figura 4.4: Distribuzione delle trasmittanze prima dell'applicazione dei filtri

percentuale molto elevata di *outlier*; la procedura ha infatti avuto come risultato l'eliminazione di circa il 60% di certificazioni dal *dataset*. Un'analisi approfondita sui certificati scartati ha fatto emergere come circa il 60% dei valori dei rendimenti di emissione, regolazione e distribuzione fosse uguale a zero; si è così supposto che questi valori potessero essere stati generati da errori dovuti alla trasmissione dei dati dai software adibiti alla redazione degli APE, errori di decodifica o errori di arrotondamento di valori molto piccoli. Del 40% di certificati rimanente sono state invece analizzate le distribuzioni dopo lo scaling, come mostrato in Figura 4.5.

Per i quattro rendimenti sono evidenziate sia la frequenza assoluta che quella cumulata; si noti come i rendimenti di emissione, regolazione e distribuzione presentino valori distribuiti in larga parte dentro i range di validità, mentre al contrario il

Figura 4.5: Distribuzioni dei rendimenti esclusi dai filtri e diversi da zero

rendimento di generazione presenti un'andamento più distribuito. Circa il 50% dei certificati si trova infatti fuori dai range di validità, con valori compresi tra 0.2 e 0.7. Si è dunque concluso che circa il 40% delle certificazioni è stato scartato a causa del rendimento di generazione. Possibili cause per questa anomalia sono da ricercare nella errata compilazione di questo specifico attributo in fase di redazione dell'APE. Valori alti di rendimento di generazione potrebbero infatti suggerire una confusione con il COP (*Coefficiente di Prestazione*) nell'eventualità della presenza di una pompa di calore come impianto di riscaldamento. Queste infatti possono raggiungere efficienze anche dell'ordine della decina, con la caratteristica però di essere ancora scarsamente diffuse in ambito residenziale.

4.2 Applicazione e Risultati delle tecniche di Data Mining

In questa fase si sono applicati, e successivamente analizzati, i risultati ottenuti attraverso l'applicazione del clustering. Questo è stato applicato su di un sottoinsieme

di attributi selezionato dall'esperto di dominio, ovvero:

- **SV** $[m^{-1}]$: fattore forma.
- **ETAH**: rendimento medio globale dell'impianto per il riscaldamento invernale.
- $\mathbf{U_t}$ $[\frac{W}{m^2K}]$: trasmittanza trasparente.
- $\mathbf{U_o}$ $[\frac{W}{m^2K}]$: trasmittanza opaca.
- $\mathbf{S_r}$ $[m^2]$: superficie riscaldata.

Questi attributi identificano un sottoinsieme di aspetti capaci di descrivere le caratteristiche termofisiche fondamentali di un'unità immobiliare. Per prima cosa le variabili sono state normalizzate attraverso la tecnica *min-max* (vedi 3.2.1), in modo da rimuovere le differenze di unità di misura e di scala presenti tra gli attributi; successivamente è stato applicato il DBSCAN per rimuovere eventuali *outlier* dalle variabili non filtrate durante l'*outlier detection domain driven*. A questo si è proceduto con l'applicazione dell'algoritmo di clustering.

4.2.1 Algoritmi di Clustering: K-Means

Gli attributi così preparati sono pronti per essere utilizzati come attributi del clustering; in particolare si è selezionato l'algoritmo K-Means. Questo, come spiegato in 3.3, necessita della definizione del parametro K, relativo al numero di cluster che fornirà in output l'algoritmo. Per la definizione ottimale di K si è ricorsi all'*Elbow Method*, plottando su di un grafico tutti i valori dell'SSE, ovvero lo scarto quadratico medio (vedi 3.3), da $K = 2$ a $K = 30$, come mostrato in Figura 4.6. Si può notare come nel nostro caso specifico il grafico non presenti un evidente *gomito* in cui scegliere K, ma piuttosto un insieme di valori che vanno da $K = 4$ a $K = 12$; in questi casi si procede sperimentalmente testando diversi valori di K compresi in questo range appena determinato. Si ricorda infatti che l'*Elbow Method* è solo un euristica

utile a determinare un punto di partenza per la determinazione del K ottimale, che ha bisogno di diverse prove sperimentali per essere affinato.

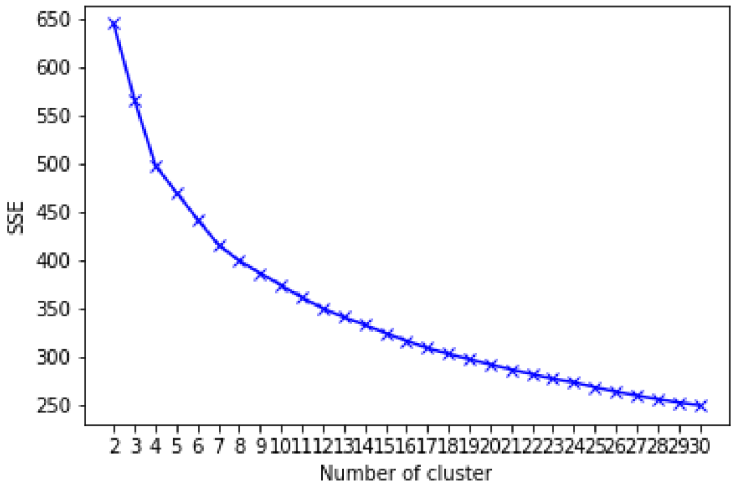


Figura 4.6: Grafico dell'SSE per valori da $K = 2$ a $K = 30$

Nel nostro caso i risultati presentati riguardano esperimenti condotti con $K = 4$, $K = 7$ e $K = 9$. Nelle figure 4.7, 4.8 e 4.9 sono visualizzati per gli attributi forniti in input al K-Means gli andamenti dei centroidi; si può notare come per tutti i vari test gli attributi più significativi sono la trasmittanza trasparente e l'ETAH. I grafici relativi a $K = 7$ e $K = 9$ mostrano inoltre come assumano importanza anche il fattore forma e la superficie riscaldata nello splitting dei cluster. Per valutare la buona riuscita del clustering si è andati ad analizzare la distribuzione dell'EP_H all'interno dei vari cluster; questa infatti fornisce un'indicazione sulla prestazione energetica degli edifici per la climatizzazione invernale.

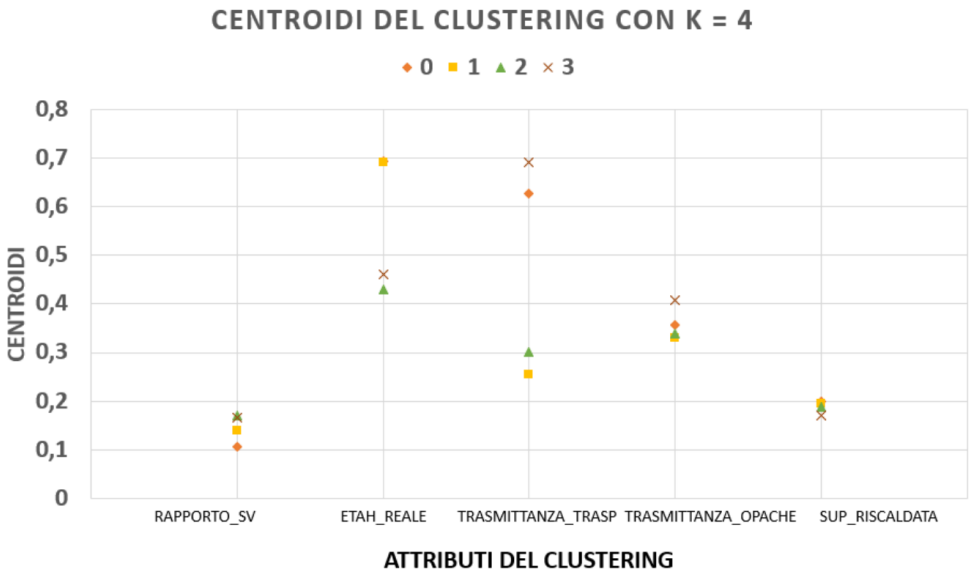


Figura 4.7: Grafico dei centroidi per $K = 4$

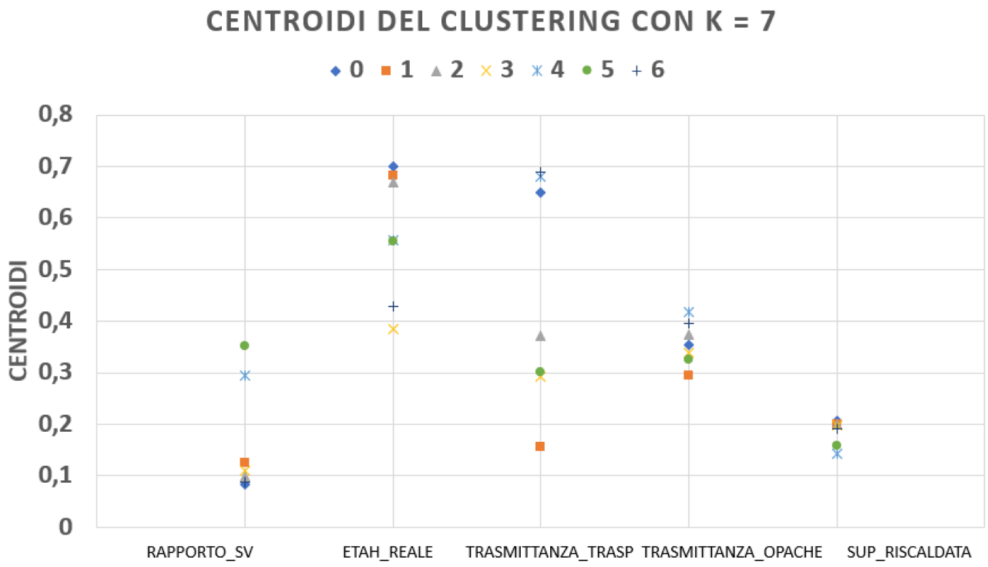


Figura 4.8: Grafico dei centroidi per $K = 7$

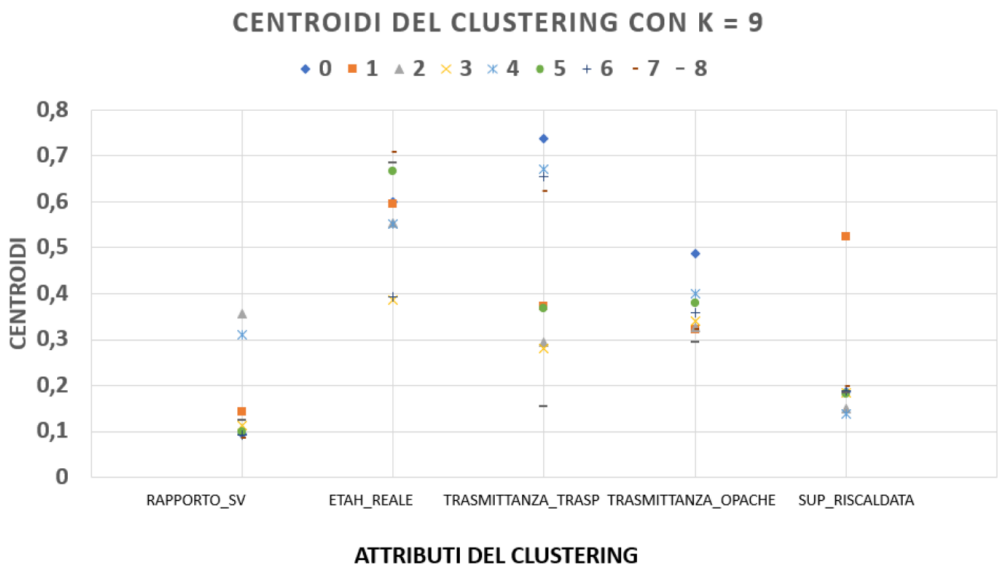


Figura 4.9: Grafico dei centroidi per $K = 9$

Con il supporto dell'esperto di dominio l'EP_H è stato diviso in tre classi:

- $0 \leq EP_H \leq 50$ alta performance $\Rightarrow$ *high*
- $50 \leq EP_H \leq 100$ media performance $\Rightarrow$ *medium*
- $EP_H \geq 100$ bassa performance $\Rightarrow$ *low*

con l'obiettivo di individuare dei cluster che riuscissero ad isolare bene edifici performanti da edifici a bassa performance. Come si nota nelle figure 4.10, 4.11 e 4.12, nonostante alcuni cluster presentino una predominanza in uno dei tre intervalli di EP_H, molti presentano la classe *medium* per circa il 40% dei certificati. Per migliorare questo risultato, si è pensato così di usare un numero elevato di K per isolare caratteristiche termofisiche ben separate, che riescano allo stesso tempo a separare comportamenti diversi di EP_H, per poi riaggregare i cluster. Per la riaggregazione si è utilizzata la distanza Euclidea calcolata sui centroidi, aggregando i cluster di quelli risultati più vicini.

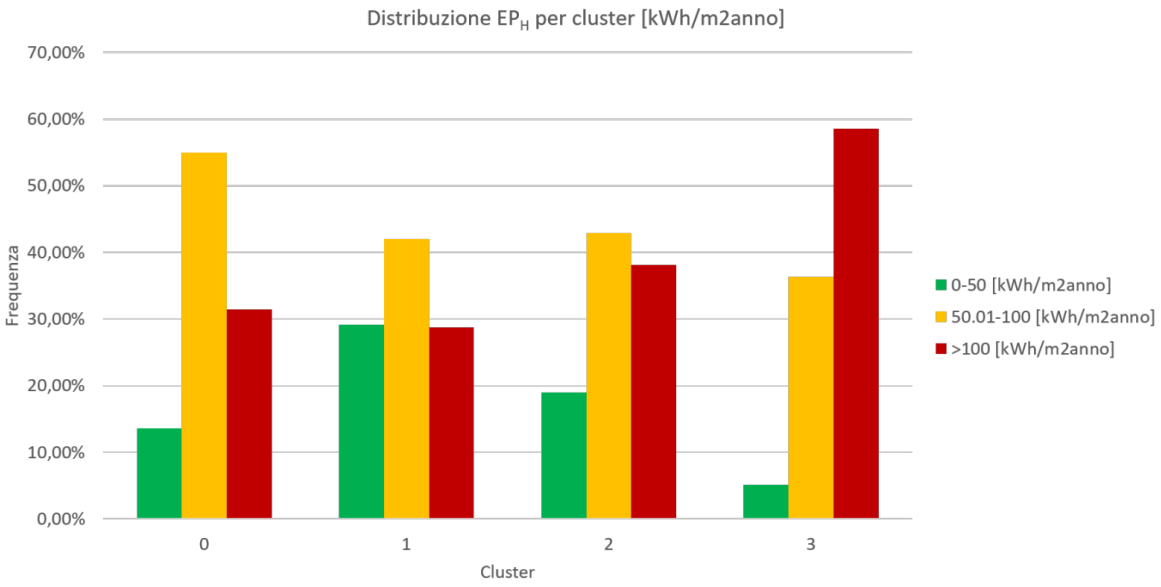


Figura 4.10: Grafico dei valori di EP_H per $K = 4$

Per migliorare l'accuratezza del processo inoltre, è stato calcolata ad ogni iterazione la *Silhouette* relativa ai cluster come media della *Silhouette* relativa ad ogni campione all'interno del cluster, in modo da avere un'indicazione sulla coesione intracluster. Questo metodo è stato sviluppato per evitare di unire cluster con centroidi vicini, ma con indici e coesioni molto diversi. Due cluster vicini sono stati così uniti solo se la loro differenza di *Silhouette* non superava una certa soglia prestabilita. Si è deciso così di partire da $K = 12$ e di riaggregare fino ad ottenere quattro cluster. Come si può notare in Figura 4.13 la cardinalità dei cluster originati dalla riaggregazione non risulta omogenea come quella originata dal K-Means con $k = 4$; i cluster con cardinalità minore sono quelli che dividono meglio le caratteristiche dell'EP_H in *high* e *low* rispetto alle performance, mentre quelli con cardinalità più grande sono caratterizzati da una composizione più equilibrata rispetto all'efficienza energetica. Il grafico dei centroidi presentato nella Figura 4.14, mostra come la trasmittanza opaca rimane l'attributo meno significativo, mentre la superficie riscaldata e il fattore forma acquistano rilevanza per il clustering. Gli attributi più significativi rimangono comunque sempre l'$ETAH$ e la trasmittanza trasparente.

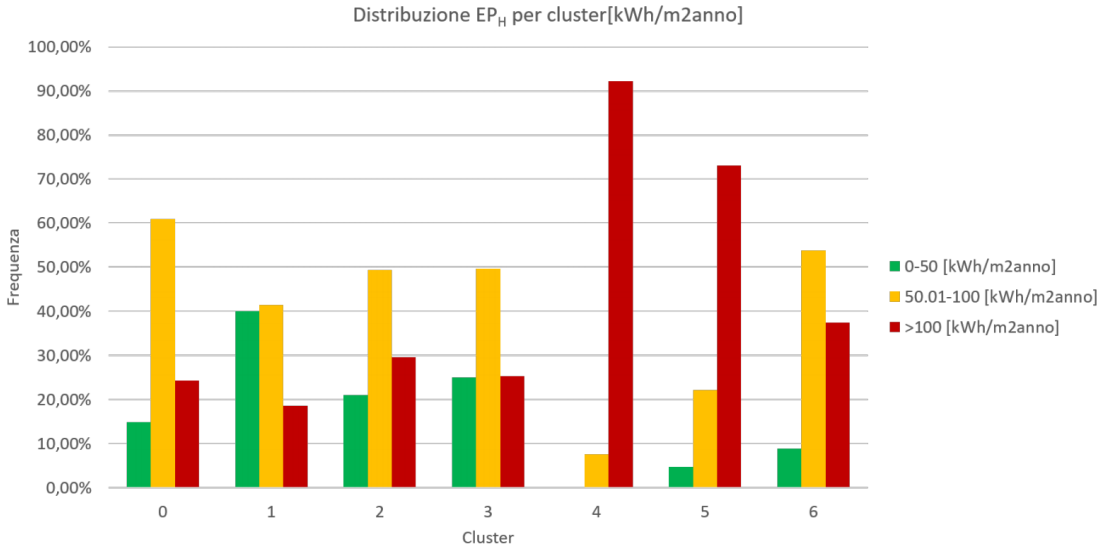


Figura 4.11: Grafico dei valori di EP_H per $K = 7$

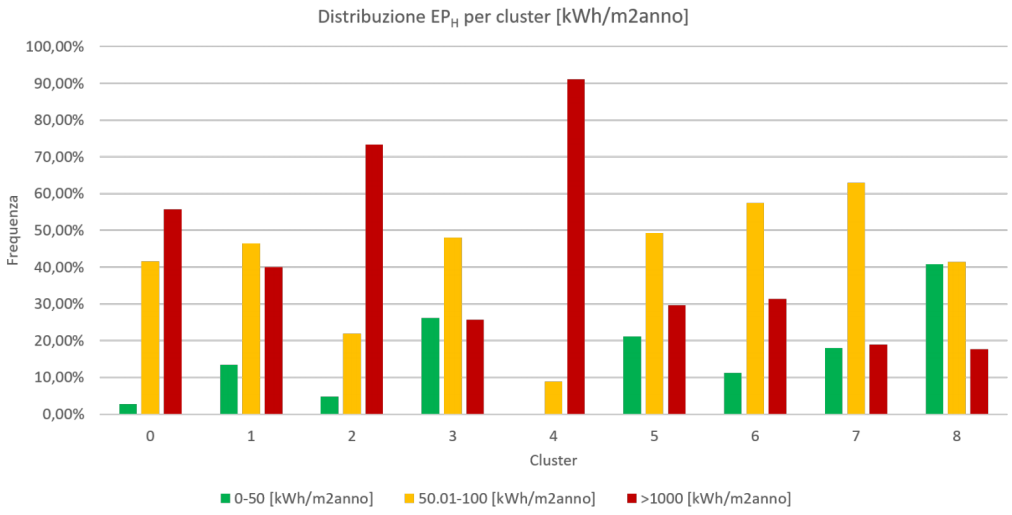


Figura 4.12: Grafico dei valori di EP_H per $K = 9$

Per analizzare più agevolmente le differenze fra i vari cluster vengono riportati i boxplot relativi ad ogni singolo attributo per cluster di appartenenza. Come si può notare dalla Figura 4.15 i valori dell'$ETAH$ si distribuiscono pressoché allo stesso

modo per i prime tre cluster, mentre sono superiori e meno dispersi nell'ultimo, evidenziando prestazioni migliori nell'impianto di riscaldamento. Dalla Figura 4.17 si può notare come la trasmittanza trasparente è l'attributo più importante nello splittare i cluster; il cluster 1 risulta quello meno performante mentre il cluster 3 risulta quello maggiormente efficiente, presentando i valori più bassi. Il fattore forma (vedi Figura 4.18) evidenzia come il cluster 2 sia quello che contiene al suo interno gli edifici meno compatti, o detto in altri termini, con i soffitti più alti.

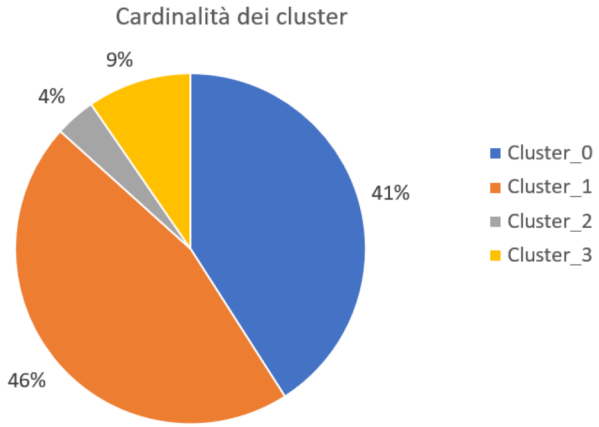


Figura 4.13: Cardinalità dei cluster riaggregati

Analizzando a questo punto per ogni cluster i relativi valori di EP_H ci si rende subito conto della coerenza dell'informazione riportata dai cluster, dove il cluster 2 appariva come uno dei meno performanti e il cluster 3 come uno dei maggiormente efficienti. Si è inoltre deciso di modificare i limiti dell'EP_H tramite un approccio maggiormente *data driven* per meglio rappresentare il reale andamento della distribuzione dell'attributo. I nuovi limiti sono stati cosi definiti come

- $0 \leq EP_H \leq 57$ alta performance $\Rightarrow$ *high*
- $57 \leq EP_H \leq 82$ media performance $\Rightarrow$ *medium*
- $EP_H \geq 82$ bassa performance $\Rightarrow$ *low*

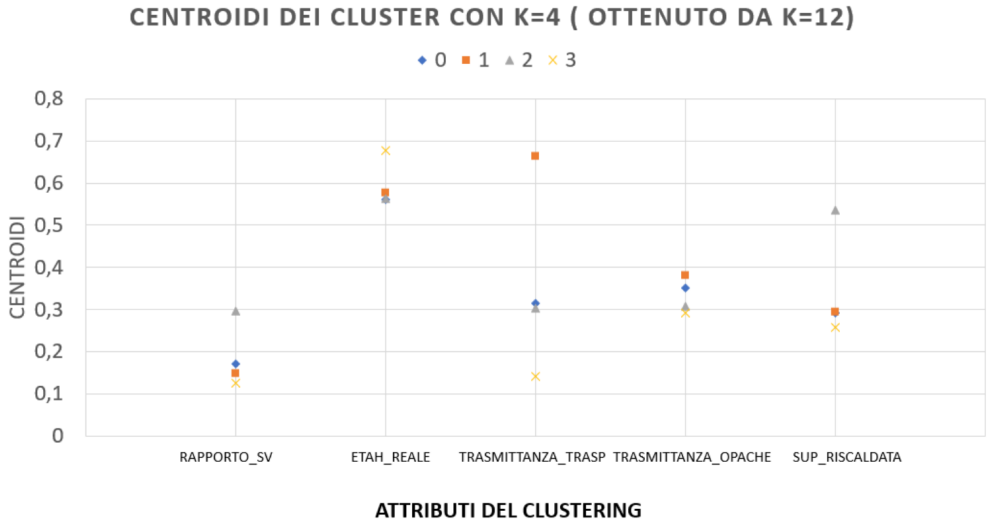


Figura 4.14: Grafico dei centroide riaggregando in 4 cluster da $k = 12$

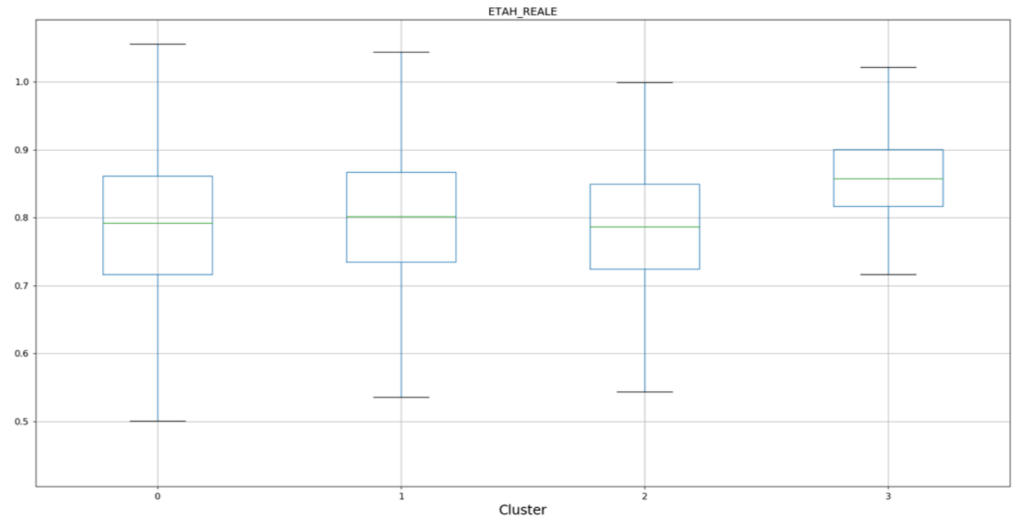


Figura 4.15: Boxplot relativo all'$ETAH$

La Figura 4.19, che conferma l'analisi tramite boxplot, mostra come il cluster 3 risulta il maggiormente performante, con una predominanza di certificati aventi EP_H appartenenti alla classe *high*, mentre il cluster 2 risulta essere il meno performante,

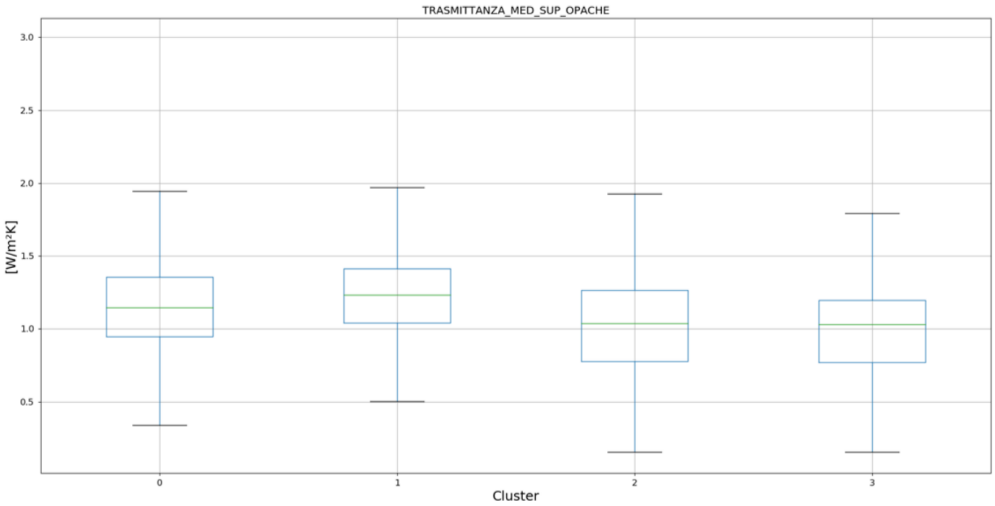


Figura 4.16: Boxplot relativo alla trasmittanza opaca

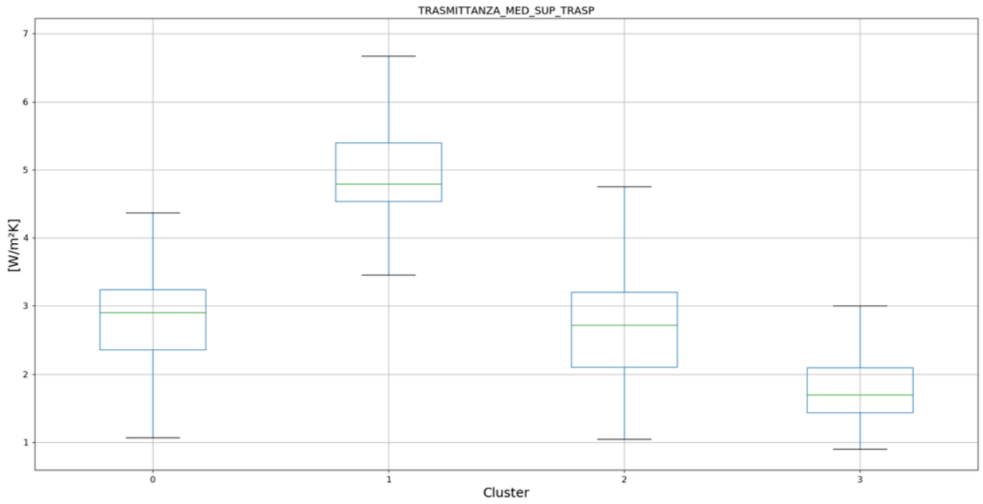


Figura 4.17: Boxplot relativo alla trasmittanza trasparente

con una predominanza di certificati aventi EP_H appartenenti alla classe *low*. Entrambi i cluster presentano una evidente diminuzione della classe *medium* rispetto

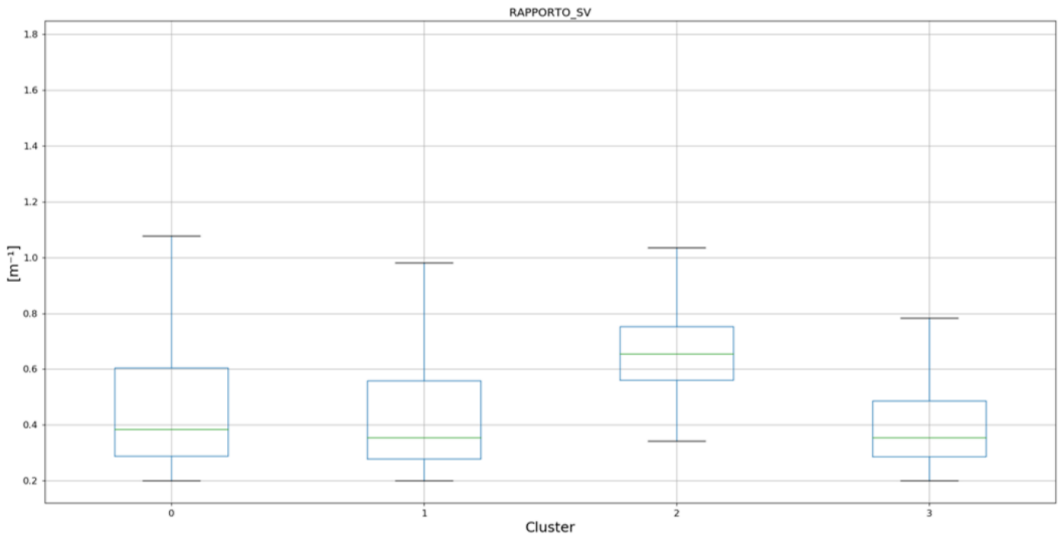


Figura 4.18: Boxplot relativo al fattore forma

al K-Means con $K = 4$, sottolineando la maggiore efficacia, in questo specifico dominio, del metodo presentato nel separare cluster con prestazioni energetiche simili. Si sono a questo punto voluti caratterizzare i cluster secondo altri attributi quali ad esempio i diversi anni di costruzione, come mostrato dalla Figura 4.20. I periodi identificati per l'analisi, in cui sono stati raggruppati i dati a nostra disposizione, sono stati ricavati da una guida rilasciata da ENEA[1] (*Agenzia nazionale per le nuove tecnologie, l'energia e lo sviluppo economico sostenibile*), e rappresentano l'entrata in vigore di leggi specifiche riguardanti l'efficientamento energetico degli edifici o determinati periodi storici. I primi due range sono relativi ai due conflitti mondiali mentre i successivi identificano:

- **1973**: emanazione di una legge responsabile di imporre limiti riguardo la dispersione termica verso l'ambiente esterno
- **1991**: emanazione della legge relativa all'attuazione del piano energetico nazionale riguardante l'uso di fonti rinnovabili

[1] http://www.enea.it/it

- **2005**: attuazione delle direttive europee in merito al rendimento energetico nell'edilizia

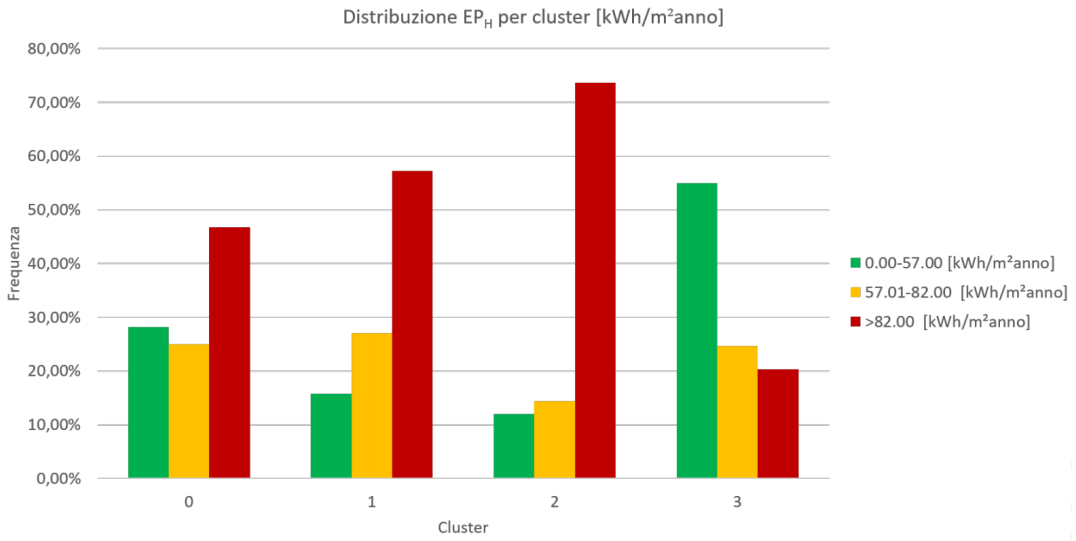


Figura 4.19: Boxplot realativo al fattore forma

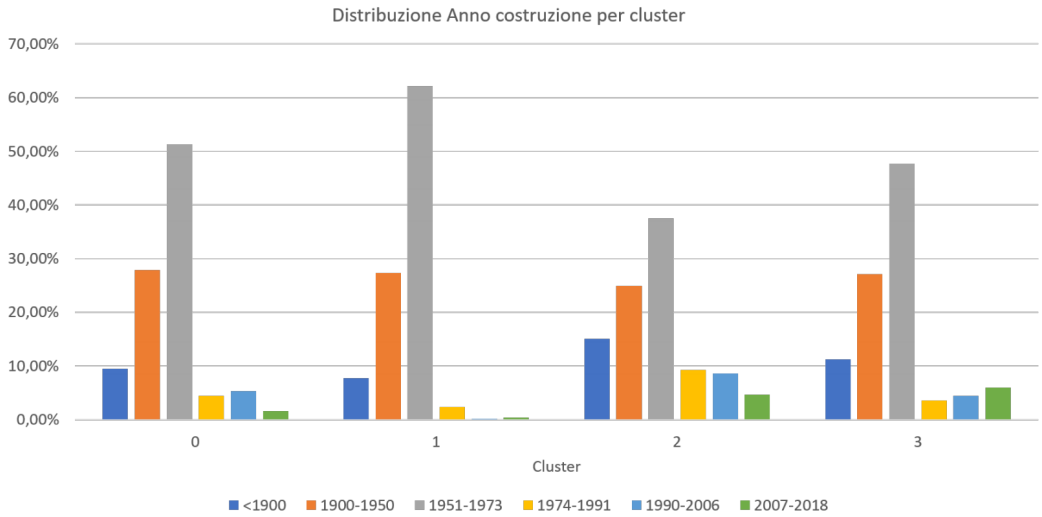


Figura 4.20: Distribuzione degli anni di costruzione nei 4 cluster dopo la riaggregazione

Come mostrato dalla Figura 4.20 circa il 60% di ogni cluster è formato da edifici costruiti prima del 1973, e ciò trova conferma nel fatto che più del 70% del nostro *dataset* iniziale presenta edifici costruiti durante gli anni '60. Questo mette inoltre ulteriormente in evidenza come sia del tutto plausibile non riscontrare grandi cambiamenti della trasmittanza opaca nei diversi cluster, dato che gli edifici appartengono pressoché allo stesso periodo storico e sono stati costruiti plausibilmente con le medesime tecniche e gli stessi materiali. Si sono a questo punto andate ad analizzare le distribuzioni delle etichette energetiche nei vari cluster, tenendo conto che le classi più performanti, ovvero A1, A2, A3 e A4 rappresentano poco più del 3% del *dataset* di partenza. Le classi energetiche sono state così accorpate in:

- A1, A2, A3, A4, B, C ⇒ alta efficienza
- D, E ⇒ media efficienza
- F, G ⇒ bassa efficienza

Come si può notare dalla Figura 4.21, dato che il *dataset* originario presenta circa il 60% di certificati con classe energetica F o G, anche nei singoli cluster queste risultano essere la maggioranza, eccezion fatta per il cluster 3, dove il massimo viene raggiunto dalla classe a *media efficienza*. Lo stesso cluster 3 presenta la più alta percentuale intracluster di edifici altamente efficienti (circa il 25%), contenendo oltre l'80% di certificati altamente efficienti sul totale di certificati a *alta efficienza* presenti nel db. Il cluster 1 al contrario emerge come quello con la più grande percentuale di cluster a *bassa efficienza* (circa l'80%). Nonostante ciò, la classe energetica non è stata considerata come affidabile nelle analisi, dato che essa viene determinata dal confronto tra l'$EP_{gl,nren}$ con il relativo valore di riferimento e che potrebbe assumere nel caso di autocertificazione il valore F, rendendo fuorviante l'informazione. Successivamente, dato che il cluster 3 sembra rappresentare la porzione di *dataset* maggiormente efficiente, si è andato ad indagare su cosa differenzi gli edifici al suo interno rispetto a quelli presenti negli altri cluster.

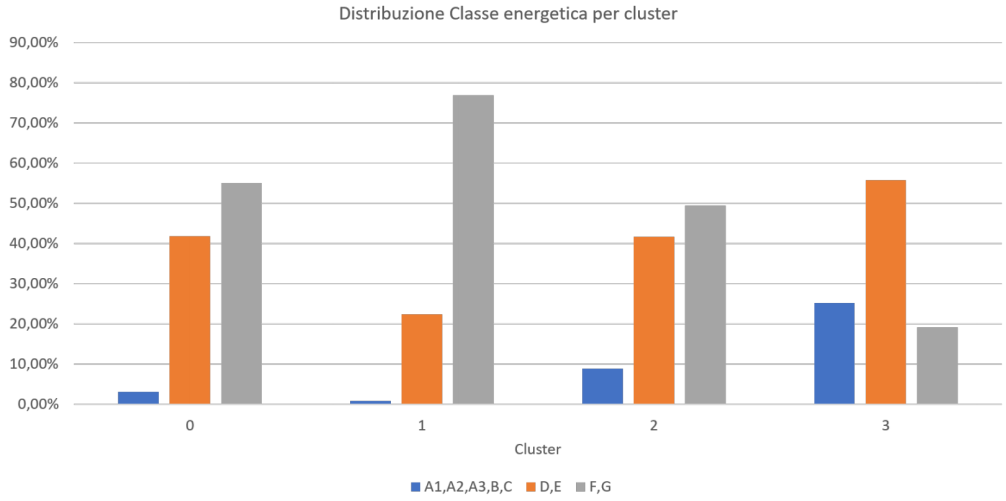


Figura 4.21: Distribuzione nei cluster delle classi energetiche aggregate

Dato che questi non si differenziano per quanto riguarda l'anno di costruzione, si è pensato di indagare sulla presenza di interventi di ristrutturazione, identificati come ristrutturazioni di primo e secondo livello o riqualificazioni energetiche. Come si può notare dalla Figura 4.22, il 20-30% circa per ogni cluster presenta l'attributo *ristrutturato* non valorizzato. Tuttavia emerge anche che il cluster 3 rappresenta l'unico cluster a registrare un percentuale consistente di abitazioni ristrutturate (circa il 30%), evidenziando come nonostante i certificati appartengano a palazzi vecchi, questi hanno subito importanti ristrutturazioni che hanno impattato sulla prestazione energetica dell'edificio. Dalla Figura 4.23 si evince come, dopo essere stati ristrutturati, il 60% dei certificati appartenenti al cluster 3 presenta un basso valore di EP_H. Al contrario tenendo conto del cluster 1, circa il 60% anche dopo la ristrutturazione continua a mantenere un alto valore di EP_H, indicando che evidentemente questa non ha avuto effetti nei riguardi dell'efficienza energetica (ampliamenti dell'immobile, riprogettazione degli spazi interni), o ne ha avuti in modo limitato.

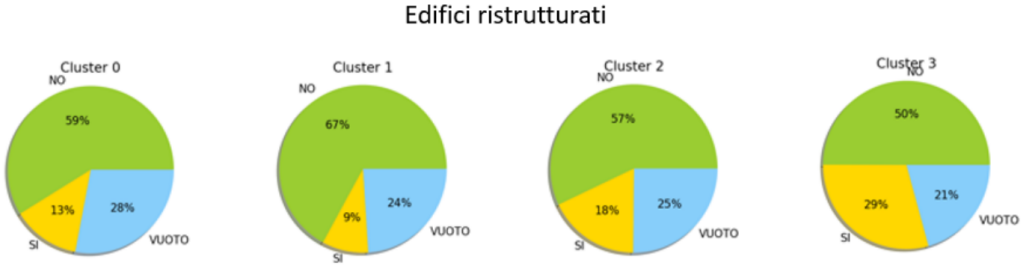


Figura 4.22: Distribuzione degli edifici ristrutturati nei 4 cluster dopo la riaggregazione

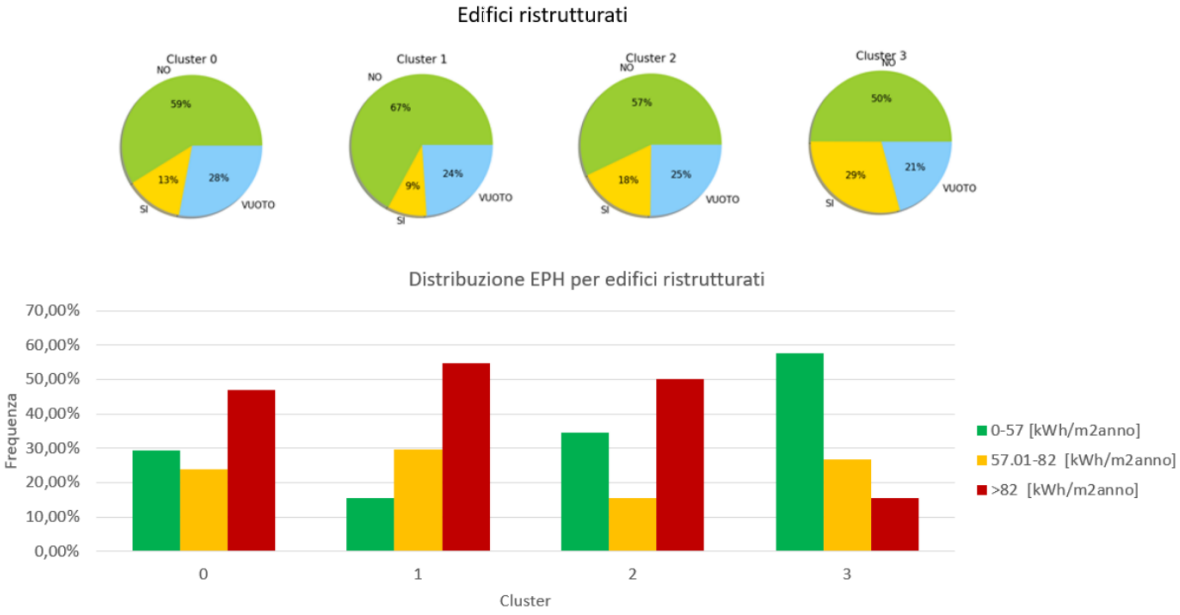
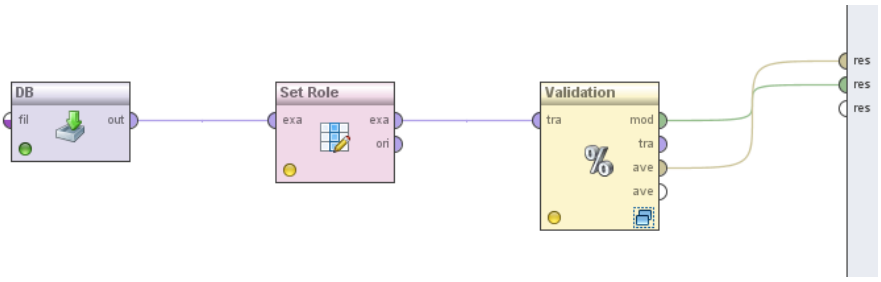


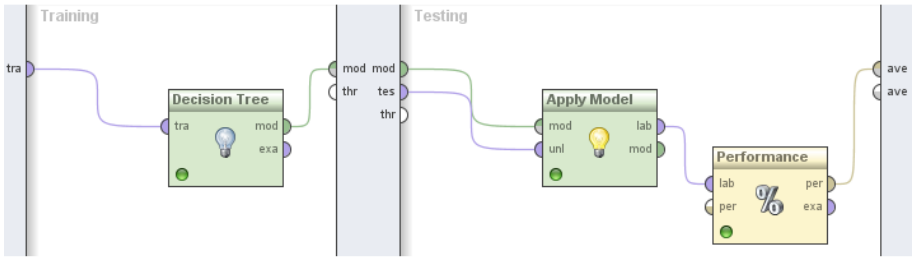
Figura 4.23: Distribuzione dell'EP_H degli edifici ristrutturati nei 4 cluster dopo la riaggregazione

4.2.2 Alberi di Decisione

Al fine di fornire una caratterizzazione migliore dei cluster ottenuti, sono stati utilizzati gli alberi di decisione, in particolare l'algoritmo C4.5. Questo è stato preferito all'ID3 perché le nostre variabili (le stesse utilizzate nel clustering) possiedono valori continui. Nella Figura 4.24 è mostrato il processo Rapid Miner attraverso il quale si è generato l'albero.



(a)



(b)

Figura 4.24: Processi di Rapid Miner per gli Alberi di Decisione

Come db in input, è stato utilizzato quello contenente le *label* originate dal clustering con $K = 12$ riaggregato in 4 gruppi. La matrice di confusione mostrata nella Figura 4.25, mostra dei risultati di *recall* e *precision* molto elevati (oltre il 90%); solo il cluster 2 ottiene dei risultati più bassi, probabilmente un effetto della bassa cardinalità. Dall'albero possiamo comunque estrarre qualche regola interessante e in linea con quello che ci aspettiamo sui cluster. Per esempio il cluster 3 viene identificato da $U_t < 2$, $ETAH > 0.8$, $S_r < 86$ e $SV < 0.6$. Questa caratterizzazione evidenzia come un cluster, e di conseguenza un edificio, per essere performante come

il cluster 3 deve avere degli infissi di qualità, un impianto di climatizzazione efficiente e dei soffitti di altezza media. Si può notare come la trasmittanza opaca non rientri nelle caratteristiche fondamentali. Questo suggerisce probabilmente, e congiuntamente all'informazione relativa alle ristrutturazioni, che non sia necessario investire in ristrutturazioni radicali dell'immobile, ma piuttosto investire in modo mirato. Consderando la trasmittanza trasparente, questa si conferma molto rilevante, posizionandosi nel nodo radice dell'albero. Si può notare come un valore di $U_t > 4$, sia la regola estratta dai cluster 2 e 1, congiuntamente ad altre caratteristiche come una grande superficie disperdente o soffitti molto alti: tutte caratteristiche di cluster molto energivori.

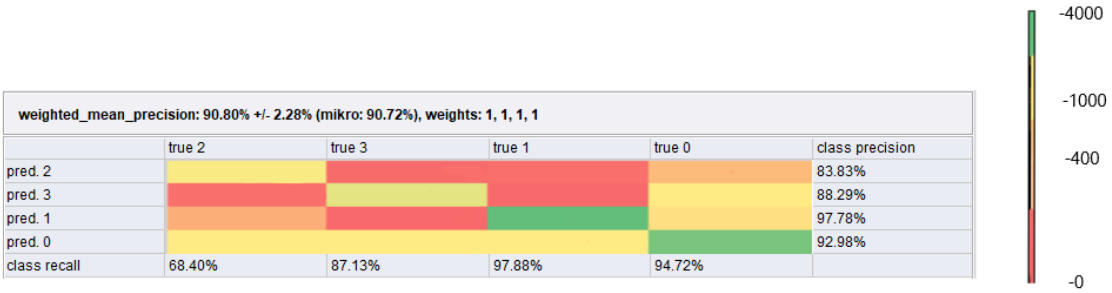


Figura 4.25: Matrice di confusione

4.3 Visualizzazione ed Interpretazione della Conoscenza

Per avere una più intuitiva ed efficace comprensione dell'informazione estratta dal *dataset* ci siamo avvalsi di mappe interattive. Durante le analisi esplorative del db, questo sono servite a visualizzare la variabile di interesse, geolocalizzando i certificati, e permettendo all'esperto di dominio di valutare l'impatto delle decisioni prese. Dopo aver applicato le tecniche di *data mining* inoltre, si sono sfruttate per visualizzare i cluster in forma aggregata, separatamente per Circoscrizione e isolato . Sono

state inoltre sviluppate delle *dashboard* dinamiche riassuntive, capaci di fornire informazioni sulle distribuzioni degli attributi relativi ai diversi cluster. Nella Figura 4.26 viene presentato un esempio di visualizzazione, prendendo in considerazione l'attributo ETAH. Si può notare come ogni isolato sia colorato in base al colore medio dei valori di ETAH contenuti al loro interno; stessa cosa può dirsi riguardo ai *marker cluster*, anch'essi colorati secondo la media dei valori, che aggregano certificati multipli. Il numero all'interno dei *marker cluster* rappresenta la cardinalità dei certificati contenuti. Si può notare anche come vengano mostrate informazioni aggregate relative alla Circoscrizione, quali le distribuzioni dell'attributo, in diverse forme grafiche per facilitarne la comprensione.

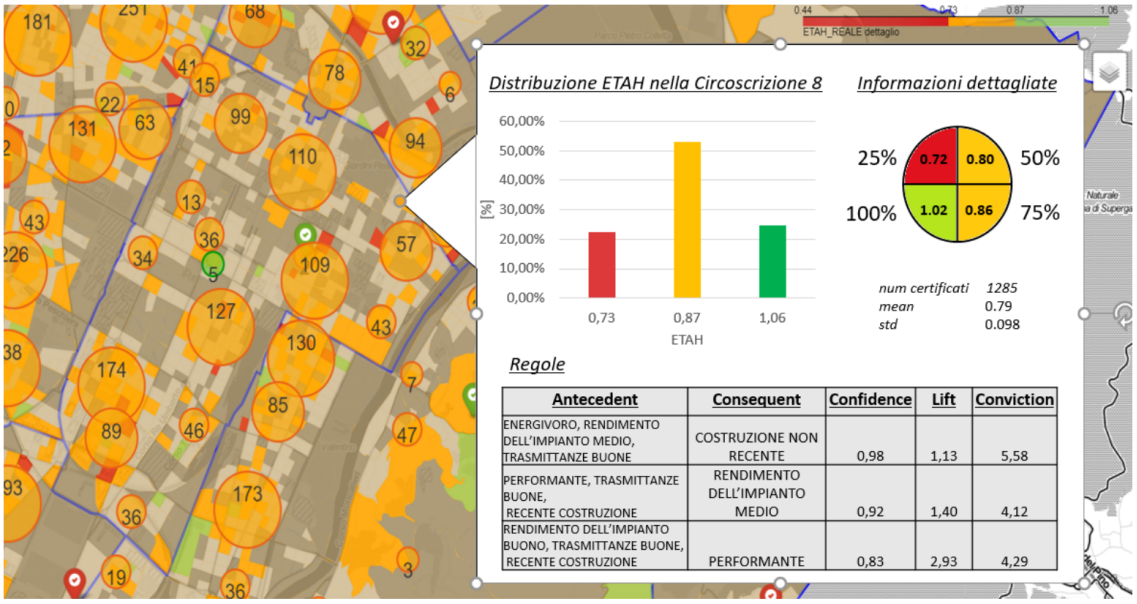


Figura 4.26: Dettaglio di *dashboard* relativo all'ETAH della Circoscrizione 8

Capitolo 5

Conclusioni e Sviluppi Futuri

L'obiettivo del presente lavoro è stato quello di progettare e sviluppare un *framework* in grado di supportare l'analista nell'estrarre conoscenza dagli attestati di certificazione energetica. Si è sviluppato il *framework* TUCANA, *tool* automatico che attraverso tecniche di *data mining* permette la caratterizzazione dell'efficienza energetica degli immobili. Nello specifico sono stati analizzati gli APE relativi agli edifici residenziali della città di Torino rilasciati dal 2016 al primo semestre 2018. Grazie all'utilizzo di TUCANA è possibile individuare e rimuovere certificati incompleti o errati, e applicare efficacemente su di un *dataset* pulito algoritmi di *data mining*. Inoltre, è possibile visualizzare la conoscenza estratta tramite il supporto di grafici e mappe geolocalizzate interattive. Queste consentono di selezionare un'area di interesse e di mostrare informazioni aggregate riguardanti la distribuzione di uno specifico attributo. La rappresentazione è stata sviluppata per risultare fruibile anche a utenti non del settore, permettendo di esplorare zone diverse della città a diversi livelli di dettaglio. La caratterizzazione delle proprietà termofisiche degli edifici, ha inoltre permesso di individuare le problematiche relativa all'efficienza energetica delle diverse aree urbane di Torino, e di proporre soluzioni specifiche.

La metodologia presentata è stata applicata esclusivamente ad edifici residenziali, ma potrebbe essere ampliata anche ad immobili con differenti destinazioni d'uso.

Al momento si stanno sperimentando nuove metodologie per il clustering (e.g., clustering di tipo gerarchico), in modo tale da raffinare la suddivisione in gruppi del *dataset*. Inoltre, si pensa di integrare i dati del catasto degli impianti termici, per costruire un modello predittivo in grado di popolare le mappe con valori stimati, laddove manchi l'informazione puntuale sul certificato.

Bibliografia

[1] Edenhofer, O., et al. *Climate Change 2014: Mitigation of Climate Change : Working Group III Contribution to the IPCC Fifth Assessment Report*. A cura di The Intergovernmental Panel on Climate Change IPCC 2014. first edition. Cambridge University Press, Cambridge, United Kingdom e New York, NY, USA, 2014. URL: www.cambridge.org/9781107654815.

[2] Capozzoli, et al. «Data mining for energy analysis of a large data set of flats». In: (2015), pp. 1–16.

[3] Basso, S. *Glossario termini Certificazione Energetica*. http://www.la-certificazione-energetica.net/glossario_4.html.

[4] Fayyad, U, Piatetsky-Shapiro, G., Smyth, P. «Knowledge Discovery and Data Mining: Towards a Unifying Framework». In: (1996), pp. 82–88. URL: http://www.aaai.org/Papers/KDD/1996/KDD96-014.pdf.

[5] Grubbs, F.E. «Procedures for detecting outlying observations in samples». In: (1969), pp. 1–21. URL: http://scholar.google.com/scholar_lookup?title=Procedures+for+detecting+outlying+observations+in+samples&author=F.+E.+Grubbs&publication_year=1969.

[6] Maddale, G. S. *Outliers*. second edition. Macmillan Publishing Company,New York: MacMillan. pp. 88–96, 1992. ISBN: 0-02-374545-2. URL: https://books.google.it/books?id=nBS3AAAAIAAJ&pg=PA89&redir_esc=y.

[7] Rosner, B. «Percentage points for a generalized ESD many-outlier procedure». In: (1983), pp. 165–172. URL: https://www.tandfonline.com/doi/abs/10.1080/00401706.1983.10487848.

[8] Grubbs, F.E. «Sample criteria for testing outlying observations». In: (1950), pp. 27–58. URL: http://scholar.google.com/scholar_lookup?title=Sample+criteria+for+testing+outlying+observations&author=F.+E.+Grubbs&publication_year=1950.

[9] Grubbs, F.E. «Extension of sample sizes and percentage points for significance tests of outlying observations». In: (1972), pp. 847–854. URL: http://scholar.google.com/scholar_lookup?title=Extension+of+sample+sizes+and+percentage+points+for+significance+tests+of+outlying+observations&author=F.+E.+Grubbs&author=G.+Beck&publication_year=1972.

[10] Adikaram, K.K.L.B. , et al. «Data Transformation Technique to Improve the Outlier Detection Power of Grubbs' Test for Data Expected to Follow Linear Relation». In: (2014), pp. 1–9. URL: http://dx.doi.org/10.1155/2015/708948.

[11] NCSS Statistical Software, cur. *Grubbs' Outlier Test*. 2018. URL: https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Grubbs_Outlier_Test.pdf.

[12] *Engineering Statistics Handbook: Detection of Outliers*. https://www.itl.nist.gov/div898/handbook/eda/section3/eda35h.htm.

[13] *Engineering Statistics Handbook: Generalized ESD Test for Outliers*. https://www.itl.nist.gov/div898/handbook/eda/section3/eda35h3.htm.

[14] Hampel, F.R. «Sample criteria for testing outlying observations». In: (1950), pp. 383–393. URL: https://www.tandfonline.com/doi/abs/10.1080/01621459.1974.10482962.

[15] Leys C., et al. «Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median». In: *Journal of Experimental Social Psychology (2013)* (2013). URL: http://dx.doi.org/10.1016/j.jesp.2013.03.013.

[16] Huber, P.J. *Grubbs' Outlier Test*. New York: John Wiley, 1981. URL: https://www.researchgate.net/profile/Mehmet_Cankaya/post/What_is_new_about_the_global_injectivity_in_terms_of_the_leading_minors_of_the_Jacobian/attachment/59d6209b79197b807797f13d/AS%

3A291621783523328%401446539459191/download/Peter_J._Huber-Robust_statistics-Wiley-Interscience%281981%29.pdf.

[17] Tukey, J.W. *Exploratory Data Analysis*. Addison-Wesley, 1977, pp. 39–49.

[18] Hubert, M., et al. «An Adjusted Boxplot for Skewed Distributions». In: (2008). URL: https://wis.kuleuven.be/stat/robust/papers/2008/adjboxplot-revision.pdf.

[19] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu. «A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise». In: (1996). URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.71.1980.

[20] Bäcklund, H., et al. *Linköpings Universitet: DBSCAN*. http://staffwww.itn.liu.se/~aidvi/courses/06/dm/Seminars2011/DBSCAN(4).pdf.

[21] Tryon, R.C. *Cluster Analysis: Correlation Profile and Orthometric (factor) Analysis for the Isolation of Unities in Mind and Personality*. A cura di lithoprinters Edwards brother Incorporated e 1939 publishers. Addison-Wesley, 1939. URL: https://books.google.it/books?id=gsnrAAAAMAAJ.

[22] Mahalanobis, Prasanta Chandra. «On the generalised distance in statistics». In: (1936), pp. 49–55. URL: https://insa.nic.in/writereaddata/UpLoadedFiles/PINSA/Vol02_1936_1_Art05.pdf.

[23] J. MacQueen. «Some methods for classification and analysis of multivariate observations». In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, Calif.: University of California Press, 1967, pp. 281–297. URL: https://projecteuclid.org/euclid.bsmsp/1200512992.

[24] Andrew W. Moore. *Decision trees: Sistemi informativi per le Decisioni*. http://www-db.deis.unibo.it/courses/SID/old/Lezioni/04%20-%20Decision%20trees.pdf.

[25] Abbas Alharan, Radhwan Alsagheer e Ali Al-Haboobi. «Popular Decision Tree Algorithms of Data Mining Techniques: A Review». In: *International Journal of Computer Science and Mobile Computing* 6 (giu. 2017), pp. 133–142.

[26] Levenshtein, Vladimir I. *"Binary codes capable of correcting deletions, insertions, and reversals"*. Soviet Physics Doklady, 1966, pp. 707–710. URL: http://adsabs.harvard.edu/abs/1966SPhD...10..707L.

[27] Luai Shalabi, Shaaban Zyad e Basil Al-Kasasbeh. «Data Mining: A Preprocessing Engine». In: *Journal of Computer Science* 2 (set. 2006). DOI: 10.3844/jcssp.2006.735.739.

[28] Steinhaus, H. «Sur la division des corps matériels en parties». In: *Bull. Acad. Polon. Sci.* (1967). URL: https://zbmath.org/?format=complete&q=an:0079.16403.

[29] Lloyd, S. P. (1957). «Least Squares Quantization in PCM». In: *Bell Telephone Laboratories Paper* (1982). URL: https://doi.org/10.1109%2FTIT.1982.1056489.

[30] Dubes, R.C., Jain, A.K. «Algorithms for Clustering Data». In: *Prentice Hall* (1988).

[31] Andrew, N. «Clustering with the K-Means Algorithm». In: *Machine Learning* (2012).

[32] Thinsungnoena, T. «The Clustering Validity with Silhouette and Sum of Squared Errors ». In: *Proceedings of the 3rd International Conference on Industrial Application Engineering 2015* (2015), pp. 44–50. URL: https://pdfs.semanticscholar.org/8785/b45c92622ebbbffee055aec198190c621b00.pdf.

[33] et al. Syakur M-A. «Integration K-Means Clustering Method and Elbow Method For Identification of The Best Customer Profile Cluster». In: *IOP Conference Series: Materials Science and Engineering* 336.1 (2018), p. 012017. URL: http://stacks.iop.org/1757-899X/336/i=1/a=012017.

[34] Pang-Ning Tan, Michael Steinbach e Vipin Kumar. *Introduction to Data Mining*. Pearson Education, 2006.

[35] Peter J. Rousseeuw. «Silhouettes: A graphical aid to the interpretation and validation of cluster analysis». In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65. ISSN: 0377-0427. DOI: https://doi.org/10.1016/0377-0427(87)90125-7. URL: http://www.sciencedirect.com/science/article/pii/0377042787901257.

[36] Friendly, M. «Milestones in the history of thematic cartography, statistical graphics, and data visualization». In: (2008).

[37] Y. Olivo, A. Hamidi e P. Ramamurthy. «Spatiotemporal variability in building energy use in New York City». In: *Energy* 141 (2017), pp. 1393–1401. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2017.11.066. URL: http://www.sciencedirect.com/science/article/pii/S0360544217319266.