



SAPIENZA
UNIVERSITÀ DI ROMA

Valutazione del danno alluvionale mediante strumenti di Statistica Multivariata e Machine Learning

Facoltà di Ingegneria dell'Informazione, Informatica e Statistica
Corso di Laurea in Statistica Gestionale

Montanari Simone

Matricola 1642957

Relatore

Prof. Maurizio Vichi

Relatori Esterni

Dr. Christian Natale Gencarelli

Dr. Simone Sterlacchini

Anno Accademico 2020/2021

**Valutazione del danno alluvionale mediante strumenti di Statistica Multivariata
e Machine Learning**

Sapienza Università di Roma

© 2022 Montanari Simone. Tutti i diritti riservati

Questa tesi è stata composta con L^AT_EX e la classe Sapthesis.

Email dell'autore: simonemontanar@gmail.com

To me

Sommario

Il presente progetto nasce da una collaborazione tra il Consiglio Nazionale delle Ricerche (CNR) e Croce Rossa Italiana (CRI).

Più precisamente, il lavoro è stato svolto sotto la supervisione del Laboratorio di Analisi dei Rischi e di Gestione delle Emergenze (LARGE) dell'Istituto di Geologia Ambientale e Geoingegneria (IGAG) del CNR di Milano, il cui compito è studiare e comprendere i processi geologici e naturali e le attività antropiche che interagiscono con l'ambiente, le attività e la vita dell'uomo.

Lo studio effettuato riguarda in particolare il fenomeno alluvionale che colpì, in data 19 gennaio 2014, i comuni di Bomporto e Bastiglia, nella provincia di Modena, provocando ingenti danni a seguito della rottura degli argini del fiume Secchia, a nord dei suddetti comuni.

Lo scopo di questo lavoro è utilizzare algoritmi di apprendimento supervisionato e non-supervisionato al fine di rendere più efficiente lo studio di questi fenomeni, analizzando le caratteristiche naturali e antropiche che incidono maggiormente su di essi.

Ad oggi esistono ancora pochi studi simili e, poichè si tratta di un fenomeno strettamente locale, non è possibile applicare modelli ricavati usando dati raccolti in seguito a fenomeni simili ottenendo risultati soddisfacenti.

Questo lavoro è stato presentato nella Sessione 4 del XV Convegno Nazionale della Sezione Geoscienze e Tecnologie Informatiche (GIT) della Società Geologica Italiana tenutosi il 20 e 21 Dicembre 2021 presso Ripatransone (AP).

Indice

1 Introduzione	5
1.1 Il Fenomeno Alluvionale	5
1.2 La Valutazione dei Danni	6
2 Approccio Metodologico e Indagine sui Dati	8
2.1 La Preparazione dei Dati	8
2.1.1 Valori Mancanti	8
2.1.2 Valori Anomali	9
2.2 Apprendimento Supervisionato	12
2.2.1 Alberi Decisionali	13
2.2.2 Random Forest	15
2.2.3 XGBoost	17
2.2.4 Interpretabilità del Modello	18
2.3 Apprendimento non Supervisionato	20
2.3.1 Clustering	22
2.4 Apprendimento per Rinforzo	28
2.5 Descrizione Dataset	29
2.5.1 I Dati dalle Schede di Danno	29
2.5.2 I Dati da Modello SWAM	32
2.5.3 I Dati da Ambiente GIS	32
3 Analisi e Risultati	34
3.1 La Preparazione dei Dati	36
3.1.1 Valori Mancanti	36
3.1.2 Valori Anomali	36
3.2 Apprendimento Supervisionato	37
3.2.1 Random Forest	38
3.2.2 XGBoost	39
3.2.3 Interpretazione e Confronto	40
3.3 Apprendimento non Supervisionato	46
4 Conclusioni e Sviluppi Futuri	51

Capitolo 1

Introduzione

1.1 Il Fenomeno Alluvionale

Un'inondazione è l'accumulo di acqua su un terreno normalmente asciutto. È causata dallo straripamento delle acque interne (come fiumi e torrenti) o delle acque di marea, o da un accumulo insolito di acqua da fonti come piogge intense o rotture di dighe o argini.

Sono molti i fattori che possono contribuire alla formazione di un'alluvione. Ci sono eventi meteorologici, come abbondanti o prolungate piogge, mareggiate, scioglimento improvviso della neve, e ci sono elementi legati all'uomo, come la gestione dei corsi d'acqua (tramite la costruzione di dighe e argini) e le modifiche apportate al terreno. L'aumento dell'urbanizzazione, per esempio, comporta nuove superfici impermeabili, altera i sistemi di drenaggio naturali e spesso porta a costruire più case nelle zone alluvionali. Nelle città, le infrastrutture sottoposte a scarsa manutenzione possono portare a inondazioni urbane.

Inoltre sono sempre più presenti fattori di inondazione legati al cambiamento climatico. A tal proposito, Pascal Peduzzi, direttore del Global Resource Information Database dello United Nations Environment Programme (UNEP) ha confermato che «while it is difficult to make a direct link between an individual extreme event and climate change, it is clear that we need to be prepared to face more intense and more frequent extreme hydro-meteorological events due to climate change».

Nel momento in cui un'alluvione colpisce una comunità, introduce una serie di potenziali conseguenze a breve e lungo termine. Le più importanti sono la perdita di vite umane e gli ingenti danni alle proprietà.

Secondo l'ultimo report pubblicato dal Centre for Research on the Epidemiology of Disasters, tra il 2001 e il 2020 sono stati registrati, in Europa, 999 disastri naturali, di cui 951 legati alla condizione climatica (di natura, quindi, meteorologica, idrologica e climatologica). Le alluvioni sono stati gli eventi più frequenti (41%), seguite da tempeste (27%), temperature estreme (23%) ed eventi secondari, quali incendi, frane, siccità (9%).

Questi eventi hanno portato circa 150.000 decessi e oltre 11 Milioni di persone danneggiate. I danni stimati ammontano a più di 200 Miliardi di Euro. Più di 6.6 Milioni di persone hanno subito danni a causa di eventi alluvionali che, sul piano economico, hanno contribuito al 50% delle perdite totali da eventi calamitosi di tipo

climatico. Questi dati concordano con quanto analizzato dall'Intergovernmental Panel on Climate Change (IPCC), secondo cui i danni causati dalle inondazioni negli ultimi 10 anni sono stati 10 volte superiori a quelli nel periodo 1960 - 1970.

Secondo il rapporto 2020 di Legambiente, in Italia dal 2010 al 2020 sono stati registrati 946 fenomeni meteorologici estremi su 507 diversi Comuni, di cui 416 sono i casi di allagamento da piogge intense mentre 118 gli eventi causati da esondazioni fluviali. Il rapporto sull'anno 2021 dell'Istituto di Ricerca per la Protezione Idrogeologica (IRPI) del CNR riporta 58 morti e 20.000 evacuati nel periodo 2016 - 2020 solamente a causa di inondazioni, individuando come regioni più a rischio per mortalità da inondazione la Valle d'Aosta, il Piemonte, la Liguria e la Sardegna.

È ormai da ritenersi consolidata la tendenza a una maggiore frequenza e intensità dei fenomeni meteorologici estremi, soprattutto in Italia in quanto al centro del Mar Mediterraneo, considerato uno degli "hot spot" del cambiamento climatico. Secondo vari studi (a esempio, Darmaraki et al., 2019) l'area del Mediterraneo sarà infatti una delle più sensibili alle conseguenze del cambiamento climatico e vedrà, negli anni a venire, rapide e ingenti precipitazioni alternate a ondate di calore.

Questi rapporti hanno il fine di chiarire cause, caratteristiche e conseguenze degli eventi naturali che si susseguono nel tempo cercando di portare l'attenzione delle Amministrazioni a un livello di prevenzione piuttosto che di gestione dell'emergenza.

1.2 La Valutazione dei Danni

La "Direttiva Alluvioni" emanata dal Parlamento Europeo (2007/60/CE) e convertita in legge dal DLgs 49/2010, ha portato la valutazione e la gestione del rischio di alluvione ad acquisire un interesse ancora maggiore, costringendo gli Stati Membri a dedicare risorse e sforzi aggiuntivi alla valutazione, mitigazione e gestione del rischio di alluvione nei più ampi contesti di possibili cambiamenti climatici, crescita demografica e cambiamenti economici.

Esistono diversi approcci per modellare i danni causati dagli eventi alluvionali: modelli sintetici e modelli empirici. I primi sono basati su analisi "what-if" per definire la relazione tra l'evento e i danni derivanti da esso ed essendo piuttosto standardizzati offrono una maggiore trasferibilità ad altre aree colpite da eventi simili. I modelli empirici, invece, si basano su dati osservati in alluvioni passate, e ciò ne riduce la trasferibilità.

Entrambi i modelli possono essere comunque implementati con strutture univariate o multivariate.

A partire dal lavoro di White del 1945, le analisi sulle perdite dovute alle alluvioni prendono in considerazione l'altezza dell'acqua come elemento principale da studiare e su cui, quindi, effettuare regressioni, trascurando altri fattori che potrebbero contribuire alla realizzazione dei danni.

Smith nel 1994 ha introdotto l'utilizzo delle curve di danno, costruite mettendo in relazione l'altezza dell'acqua, sulle ascisse, con il danno, sulle ordinate. Ad oggi le curve di danno sono lo strumento adottato a livello internazionale per la stima degli impatti diretti sugli edifici residenziali. Come riporta Molinari et al. (2012), «la loro affidabilità è discutibile. I modelli esistenti [...] considerano un limitato numero di variabili esplicative (in genere l'altezza di allagamento e qualche parametro

di vulnerabilità). Al contrario, l'impatto generato dall'evento dipende da una molteplicità di fattori legati tanto alla pericolosità (es. altezza di allagamento, velocità dell'acqua, presenza di sedimenti) quanto alla vulnerabilità (numero di piani, stato di manutenzione, presenza di impianti, ecc.)».

Nasce quindi la necessità di sviluppare modelli di danno più efficaci e affidabili, inclusivi delle molteplici variabili concorrenti alla realizzazione del danno.

Grazie anche all'avanzamento tecnologico, oggi è possibile sfruttare metodi più complessi e includere nei modelli maggiori variabili, come la velocità dell'acqua, la durata dell'evento, la presenza e la tipologia di misure precauzionali. È stato inoltre dimostrato come questi modelli superino, in validità, i modelli univariati, a condizione che vengano forniti dataset sufficientemente ampi e dettagliati (Merz et al., 2013; Schröter et al., 2016).

In letteratura sono presenti diversi studi che applicano metodologie multivariate (ad esempio Hasanzadeh et al., 2016, e Kreibich, 2017) ma è stato mostrato come i modelli sviluppati all'estero non si adattino al territorio italiano e quelli applicati in alcune zone d'Italia non restituiscano risultati soddisfacenti in altre aree (Amadio et al., 2016; Molinari et al., 2012). Molinari et al. (2012) associa le basse performance nell'utilizzo di modelli già sviluppati a due principali ragioni: l'estrema variabilità geomorfologica dell'Italia accompagnata da una non standardizzazione delle tipologie edilizie e dataset italiani di bassa qualità, spesso relativi a piccole aree, a differenza degli studi europei.

Parallelamente allo sviluppo di nuovi modelli, nasce quindi anche l'esigenza di strutturare metodologie atte a raccogliere e conservare in modo adeguato i dati misurati durante e dopo questa tipologia di eventi.

È nello studio di nuovi modelli di analisi che trova spazio il lavoro descritto nei prossimi capitoli. Nel dettaglio:

- nel Capitolo 2 vengono spiegate le metodologie utilizzate e descritta la raccolta dei dati;
- nel Capitolo 3 sono presentate le analisi svolte e commentati i risultati ottenuti;
- nel Capitolo 4 sono riportate osservazioni conclusive e suggerimenti per sviluppi futuri.

Capitolo 2

Approccio Metodologico e Indagine sui Dati

In questo capitolo sono presentati e descritti gli algoritmi e le principali analisi statistiche adottate nelle varie fasi dello studio. È inoltre descritto l'evento alluvionale oggetto di studio e la struttura del dataset a disposizione. Saranno presenti anche rappresentazioni grafiche come supporto per una più chiara comprensione di quanto descritto.

2.1 La Preparazione dei Dati

In ogni lavoro statistico, la fase più importante è la preparazione dei dati (*data cleaning*) in quanto da questa deriva la bontà dei risultati finali delle analisi.

Elementi principali di questa fase sono la correzione ortografica e semantica dei dati testuali, la gestione dei valori mancanti e di quelli anomali.

La prima può essere evitata con un'efficace raccolta dei dati seguita da una adeguata gestione di quest'ultimi. Le altre sono affrontate brevemente di seguito.

2.1.1 Valori Mancanti

I valori mancanti, chiamati anche *missing values* o *not available* (*NA*), sono un problema ricorrente in molte analisi e, nonostante le attenzioni che si possono prendere in fase di raccolta dei dati, è difficile prevenire questo fenomeno. Ci sono, però, vari metodi per trattarli.

Seguirà una breve trattazione a scopo informativo mentre per una discussione più approfondita si rimanda a Graham & Hofer (2000) e Graham et al. (2003).

Anzitutto è bene specificare che esistono due scenari possibili di non risposta: mancata risposta totale o parziale.

È inoltre sempre utile chiedersi cosa si nasconde dietro la mancata risposta.

Graham et al. (2003) scrivono che i dati mancanti sono causati da una combinazione di tre elementi: fattori casuali, fenomeni misurati e fenomeni non misurati.

A questa descrizione si aggiunge quella relativa al meccanismo di non risposta elaborata da Little & Rubin (1987), che fanno ricadere quest'ultimo in una delle seguenti tre categorie:

1. *Missing Completely at Random* (MCAR): il meccanismo di non risposta è indipendente dalle variabili rilevate e si possono quindi assumere come dati casualmente mancanti. In questo caso si eliminano le non risposte senza rischiare di incorrere in bias. Conseguentemente ci sarà una ridotta potenza nelle analisi che si andranno a svolgere successivamente;
2. *Missing at Random* (MAR): in questo caso la mancanza di dati è dovuto alle variabili osservate che riportano dati mancanti. Il meccanismo di risposta è ancora ignorabile se nelle analisi si tiene conto dei fattori che hanno causato la mancanza;
3. *Not Missing at Random* (NMAR): il meccanismo dipende da fattori non analizzati dal ricercatore. Questo è il caso più complicato, e la soluzione migliore risiede nel cercare di avvicinarsi al sistema MAR tramite la costruzione di un modello che tenga in considerazione più variabili possibili.

Una volta capita la natura del meccanismo di non risposta, si può procedere con il trattamento dei dati mancanti. Esistono tre principali strategie:

- Complete-Case Analysis: si escludono dal dataset tutti i dati mancanti;
- Imputazione Semplice: se una sola variabile presenta dati mancanti, questi possono essere imputati. In questo caso ci sono varie tecniche, le più famose sono l'uso della regressione e la sostituzione dei valori mancanti con il valore medio o mediano. Ciascuno dei due sistemi ha pro e contro quindi la strategia da adottare va scelta in base al singolo problema;
- Imputazione Multipla: come l'imputazione semplice ma estesa ad altre variabili.

Per approfondire gli aspetti teorici e pratici, si rimanda a Wayman (2003), Camero & Trivedi (2005) e Soley-Bori (2013).

2.1.2 Valori Anomali

La definizione di cosa sia un valore anomalo, o outlier, non è univoca. Nel 1969 Grubbs descriveva un outlier come una «*osservazione che sembra discostarsi di molto dagli altri valori della popolazione*». Hawkins nel 1980 amplia questa definizione introducendo il meccanismo di generazione dei dati: «*un outlier è un'osservazione che si discosta così tanto dalle altre osservazioni da destare il sospetto che sia stata generata da un diverso meccanismo*». Questa formulazione ha quindi una base prettamente statistica, assumendo che i dati “normali” siano frutto di un meccanismo, ad esempio un preciso processo statistico, mentre i valori anomali deviano da tale meccanismo di generazione.

Nel 1994 Barnett & Lewis riassumono queste definizioni affermando che un valore anomalo è «*un'osservazione che sembra essere incoerente con il resto di quella serie di dati*

.

Anche questo, come i valori mancanti, è un problema molto frequente, ed è quindi necessario trattarlo con consapevolezza.

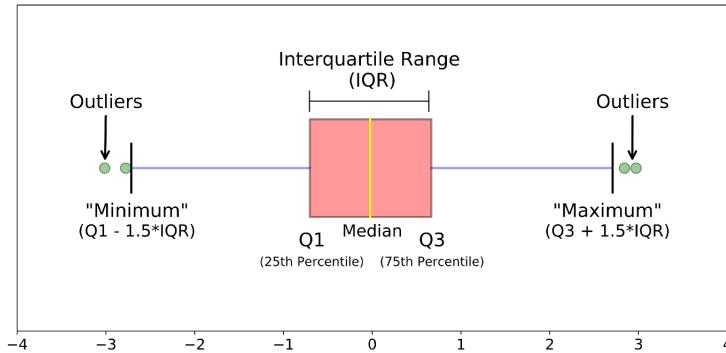


Figura 2.1. Boxplot ([74])

Ci sono casi in cui avere degli outlier è fondamentale per fare delle analisi specifiche su questi, come, ad esempio, nel rilevamento delle frodi dove, tra tante transazioni corrette, bisogna individuare quelle anomale, nelle applicazioni in campo sanitario, in cui sintomi insoliti possono indicare potenziali rischi di salute per il paziente, o nello sport, dove vengono registrate le performance degli atleti e tramite gli outlier si possono individuare quelli più promettenti.

Ben-Gal (2005) evidenzia come i metodi per rilevare gli outlier si dividono in univariati e multivariati, a seconda che si vogliano trovare i valori anomali di una singola variabile o di più variabili contemporaneamente (outlier univariati non necessariamente lo sono anche in uno spazio n-dimensionale, e viceversa).

Seguiranno ora alcuni accenni ai metodi di rilevazione dei valori anomali. Per maggiori dettagli si rimanda a Barnett & Lewis (1994), Gnanadesikan (1997) e Johnson & Wichern (2007).

Per individuare gli outlier univariati, il metodo ad oggi più diffuso per individuarli prevede l'uso del range interquantilico e di un boxplot come supporto.

A titolo esemplificativo esaminiamo il boxplot in Figura 2.1. In rosa viene evidenziato il range interquartile (IQR) calcolato come $IQR = Q3 - Q1$, con Q1 e Q3 rispettivamente primo e terzo quartile. Per selezionare i dati outlier è definito un nuovo range, in violetto, il cui limite inferiore (LI) e superiore (LS) è dato da, rispettivamente, $LI = Q1 - 1.5 \times IQR$ e $LS = Q3 + 1.5 \times IQR$ ¹. Tutti i valori esterni a questo range sono considerati anomali.

Altri due metodi diffusi ma meno usati sono lo z-score e i test d'ipotesi.

Mentre il procedimento univariato è standardizzato, nel caso multivariato la situazione è più complicata e richiede maggiori attenzioni. La complessità risiede nella scelta dell'algoritmo da utilizzare in quanto si possono ottenere output molto diversi. In Figura 2.2 sono rappresentati alcuni dei metodi maggiormente utilizzati:

- *Isolation Forest*: metodo *ensemble* derivante dall'algoritmo *Random Forest* e basato, quindi, su alberi di decisione. Viene prima scelta casualmente una variabile e poi selezionato un valore casuale, compreso tra il valore minimo e massimo della caratteristica scelta, per iniziare a suddividere le osservazioni. Il

¹Per approfondire la formula si veda <https://towardsdatascience.com/why-1-5-in-iqr-method-of-outlier-detection-5d07fdc82097>

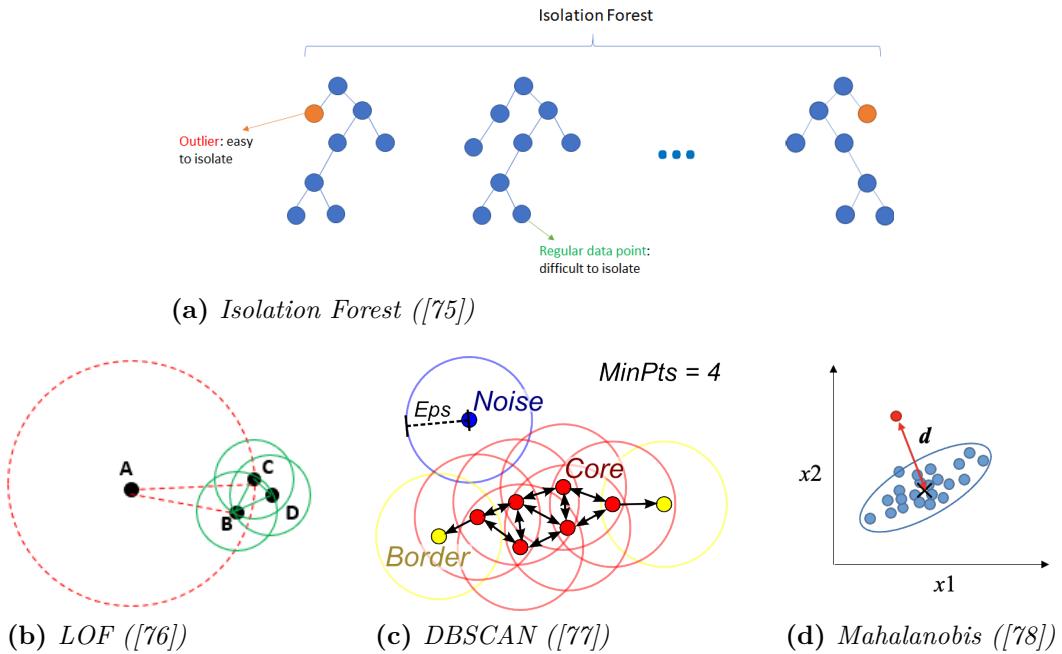


Figura 2.2. Valori anomali multivariati

processo continua finchè non viene isolata ogni singola unità o non è raggiunta una specifica profondità. Gli outlier sono meno frequenti e con valori lontani dalle osservazioni regolari. A tutte le foglie viene assegnato un punteggio, calcolato come $s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}$, in base al quale si può affermare se quel valore è anomalo o meno (Liu et al., 2008);

- *Local Outlier Factor (LOF)*: fissato un valore k , l'algoritmo analizza la densità di ciascuna unità rispetto ai k punti più vicini e assegna così un punteggio all'osservazione esaminata. Basandosi sulla densità, più aumenta la distanza dai k vicini, più aumenta l'area intorno al punto in esame, più sono radi i punti all'interno dell'area. Un valore LOF alto ($LOF \gg 1$) indica quindi che la densità del punto è bassa mentre quella dei k vicini è alta (e quindi questi k vicini sono poco distanti tra loro) e pertanto può essere considerato un outlier (Breunig et al., 2000). Esistono anche delle sue varianti ma il procedimento logico alla base rimane il medesimo;
- *DBSCAN*: è un classico algoritmo di clustering. Seleziona casualmente un punto non appartenente ad alcun cluster e determina se è un punto centrale di un possibile nuovo cluster controllando se ci sono almeno n punti entro una distanza pari a ε . Se intorno a questi esistono altri punti a una distanza ε , vengono aggiunti al cluster. In questo modo rimangono isolate quelle osservazioni che hanno intorno un numero di unità inferiore a n , che sono dunque outlier (Ester et al., 1996; Thang & Kim, 2011; Behera & Rani, 2016);
- *Distanza di Mahalanobis*: individua gli outlier osservando la distribuzione dei dati. È molto utilizzata in quanto, facendo uso della matrice delle co-

varianze, restituisce buoni risultati quando le variabili sono molto correlate e/o hanno differenti scale di misurazione. Viene calcolata la distanza tra un punto e il centro della distribuzione (si considera la media), dopodichè vengono considerati anomali tutti i valori al di fuori dell'ellisse avente raggio pari a $\sqrt{\chi^2_{p,0.95}}$ o $\sqrt{\chi^2_{p,0.99}}$. La distanza è calcolata, per un dataset $n \times p$, come $MD_i := ((x_i - t)' C^{-1} (x_i - t))^{1/2}$ per $i = 1, \dots, n$, dove t è il vettore delle medie e C la matrice di varianze e covarianze (Filzmoser, 2004; Ben-Gal, 2005; Filzmoser et al., 2005; Warren et al., 2011).

Una volta completata la fase di data cleaning, si procede visionando più nel dettaglio i dati raccolti.

Generalmente è utile iniziare applicando le analisi appartenenti alla statistica descrittiva e quindi individuare la distribuzione dei dati tramite grafici, indici di misura e variabilità, nonché indici di relazione tra le variabili. Così facendo, si riesce ad avere una prima idea del fenomeno oggetto di studio, individuarne eventuali peculiarità e selezionare le analisi più adeguate allo stesso.

2.2 Apprendimento Supervisionato

Per sviluppare questo progetto si è fatto ricorso ad algoritmi di machine learning, seguendo quanto riportato in Merz et al. (2013), Chinh et al. (2016), Carisi et al. (2018).

Ad oggi esistono tre categorie di algoritmi di machine learning: ad apprendimento supervisionato, ad apprendimento non supervisionato, ad apprendimento per rinforzo.

Sarà data una breve descrizione di tutte e tre le tipologie, approfondendo le particolarità utili a comprendere al meglio il lavoro descritto in questo testo. Per approfondimenti si veda Hastie et al. (2017), Géron (2019), James et al. (2021).

Data una matrice di dati $n \times m$ (n osservazioni, m variabili), definiamo con X_i , $i = 1..m - 1$, le variabili di input. Sia Y , quindi, la m -esima variabile, detta output. Le prime influenzano in qualche modo la seconda e lo scopo dell'apprendimento supervisionato è usare gli input per predire il valore dell'output.

Utilizzando una terminologia statistica, gli input sono detti predittori, variabili indipendenti o features.

L'output, invece, è chiamato risposta o variabile dipendente.

Un algoritmo che opera su un dataset così fatto è detto di apprendimento supervisionato (*supervised learning*) in quanto il suo scopo è imparare dagli input e output dati (fase di addestramento o training) e creare un modello predittivo con cui, dati dei nuovi input, ottenere un output non osservato \hat{y} . In una forma elementare, un algoritmo di apprendimento supervisionato può essere scritto come

$$\hat{y} = f(x_1, \dots, x_n). \quad (2.1)$$

Al fine di ottenere un modello performante anche con nuovi dati, il dataset iniziale viene suddiviso in, almeno, due parti, chiamate training set e test set.

La prima, in cui ricadono, di solito, la maggior parte delle osservazioni, ha lo scopo di allenare l'algoritmo più volte tramite, anche, la tecnica della cross validation così da renderlo teoricamente robusto a nuovi input. Questa è infatti una tecnica statistica che permette di usare i dati in modo alternato sia per la fase di train che di test. Molto utilizzata è la k -fold cross validation in cui il dataset iniziale viene suddiviso in k parti uguali e, iterativamente, la k -esima parte costituisce il test set mentre le restanti formano il train set. Il risultato finale è una media delle performance delle varie iterazioni.

È in questa fase che viene generata la funzione f dell'equazione 2.1.

Una volta “allenato”, il modello viene testato sul dataset di test che, per come è creato, contiene dati nuovi per l'algoritmo. Utilizzando quindi delle misure di performance, si riesce a trovare il modello più adatto al problema.

In base alla natura della variabile dipendente, questi algoritmi possono essere suddivisi in due sottocategorie:

- Classificazione: y è una variabile qualitativa (o, equivalentemente, categorica) quindi l'algoritmo dovrà assegnare i nuovi input a delle classi. Gli algoritmi di classificazione più popolari sono:
 - Regressione Logistica (*logistic regression*)
 - Macchine a Vettori di Supporto (*support vector machine*)
 - Classificatore Naïve Bayes (*Naïve Bayes classifier*)
 - Alberi Decisionali (*decision trees*)
 - Foresta Casuale (*random forest*)
 - Reti Neurali (*neural networks*)
- Regressione: y è una variabile quantitativa quindi l'output sarà un valore numerico. Gli algoritmi più popolari sono:
 - Regressione Lineare (*linear regression*)
 - Alberi Decisionali (*decision trees*)
 - Foresta Casuale (*random forests*)
 - Alberi Decisionali Potenziati (*boosted decision trees*)

Il caso studio rientra nella sottocategoria della regressione.

2.2.1 Alberi Decisionali

I modelli di regressione lineare falliscono nel caso in cui la relazione tra input e output non è lineare o esiste una correlazione tra le variabili. In questi casi un primo approccio all'analisi dei dati avviene attraverso gli alberi decisionali.

Questi modelli partizionano lo spazio delle variabili in sottoinsiemi sempre più piccoli fissando, di volta in volta, dei valori di cutoff. In questo modo ogni istanza apparterrà a uno e uno solo di questi sottoinsiemi.

Si ottiene così una rappresentazione che ricorda un albero e, per questo, il sottogruppo in cima è detto nodo radice, i sottoinsiemi finali sono detti nodi terminali,

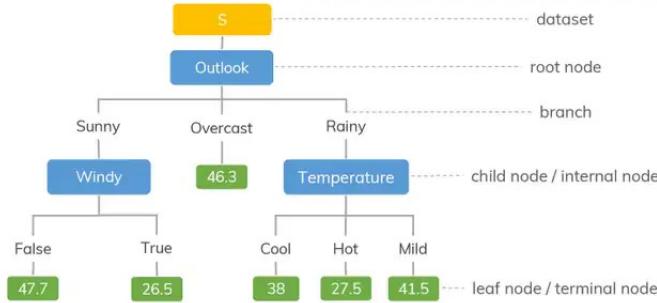


Figura 2.3. Albero Decisionale ([79])

o foglie, mentre i sottoinsiemi intermedi sono chiamati nodi interni. Le connessioni tra i nodi sono chiamate rami.

Esistono vari meccanismi di generazione dell'albero ma il più diffuso è il *Classification And Regression Trees (CART)*, proposto da Breiman et al. (1984). Elementi fondamentali in questa tipologia di algoritmi sono il partizionamento e la profondità.

Partizionamento

Un albero decisionale, quindi, divide iterativamente i dati di training in sottogruppi fino a ottenere le foglie, a cui assegna una costante (di solito la media degli output appartenenti a quel nodo).

Questa suddivisione segue una partizione binaria creata mettendo in relazione le modalità delle variabili osservate e il cutoff. L'iterazione continua finché non viene soddisfatto il criterio scelto per l'arresto.

L'output prodotto dal modello è basato sui valori medi delle modalità espresse dalle osservazioni che ricadono in quel sottogruppo.

Per una migliore comprensione si veda la Figura 2.3.

La funzione che descrive la relazione tra l'output predetto \hat{y} e le variabili x è la seguente:

$$\hat{y} = f(x) = \sum_{g=1}^G c_g I_{\{x \in R_g\}} \quad (2.2)$$

Ogni record appartiene esattamente a una foglia (corrispondente al sottoinsieme R_g). $I_{\{x \in R_g\}}$ è la funzione d'identità che vale 1 se $x \in R_g$ e 0 altrimenti, mentre c_g è la costante.

L'obiettivo di ogni nodo è trovare la variabile x_i con cui partizionare i restanti dati in due regioni, R_1 e R_2 , in modo da minimizzare l'errore tra la variabile risposta osservata y_i e il valore predetto per quella regione, c_i .

Nel caso della regressione, la minimizzazione riguarda il totale delle somme dei quadrati degli errori (*Sum of Squares Error, SSE*) ed è definita come:

$$SSE = \sum_{i \in R_1} (y_i - c_1)^2 + \sum_{i \in R_2} (y_i - c_2)^2 \quad (2.3)$$

Questa tipologia di partizionamento si ripete, come accennato poco sopra, su ogni nuova regione fino a quando non viene raggiunta una regola di arresto. È

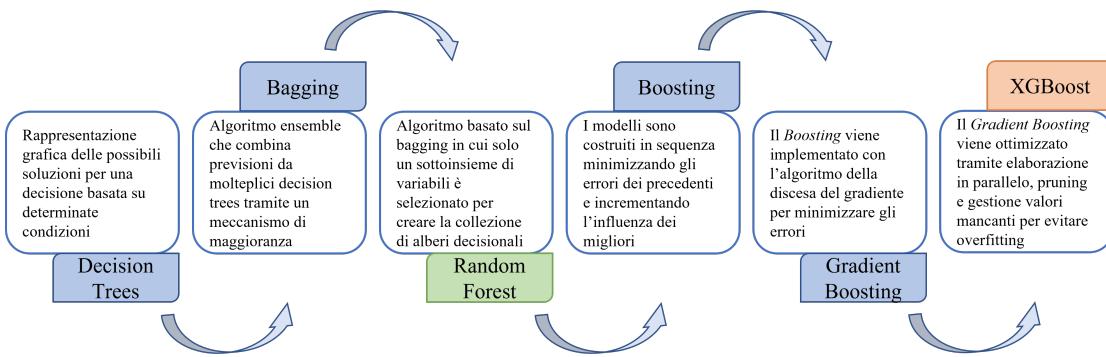


Figura 2.4. Evoluzione algoritmi ad albero

importante sottolineare che una singola variabile può essere utilizzata più volte per suddividere i dati se risulta essere quella che restituisce la partizione migliore.

Profondità

Questo tipo di alberi possono essere formati da un elevato numero di partizioni. Così facendo, però, risulterebbero troppo complessi portando, inoltre, all'overfitting del training set con il risultato di pessime prestazioni sul test set.

È necessario, quindi, trovare una sorta di equilibrio tra profondità e complessità dell'albero per ottenere previsioni soddisfacenti su nuovi dati.

Per trovare questo equilibrio si utilizzano di solito tre approcci:

1. si imposta una profondità massima, intesa come distanza massima tra la radice e una foglia (*max_depth*);
2. si imposta un numero minimo di osservazioni che devono essere presenti in una foglia così da partizionare solo i nodi contenenti almeno un certo numero di dati (*min_sample_leaf*);
3. si riduce la dimensione dell'albero rimuovendo i nodi che non forniscono informazioni aggiuntive, senza quindi ridurne l'accuratezza predittiva (*pruning*).

Vediamo ora nel dettaglio gli algoritmi utilizzati nello studio: *random forest* e *xgboost*, evoluzioni dei decision trees (Figura 2.4).

2.2.2 Random Forest

Prima di parlare di foreste casuali, è necessario introdurre due concetti: il metodo *ensemble* e il *bagging*.

Con ensemble si intende qualsiasi algoritmo che utilizza contemporaneamente più modelli di apprendimento per migliorare la qualità del risultato che si otterrebbe dall'utilizzare invece un singolo modello.

Il bagging (abbreviazione di *bootstrap aggregation*) è un esempio di metodo ensemble utilizzato per ridurre la varianza, evitare l'overfitting e migliorare l'accuratezza di un algoritmo di machine learning.

Tramite questo metodo vengono generati dei dataset cosiddetti *bootstrapped*, ossia dei sottoinsiemi creati con un campionamento casuale semplice con ripetizione, di

Algoritmo 1 Random Forest

1. Dato un training set
 2. Indicato il numero di alberi da costruire (n_trees)
 3. Per $i = 1..n_trees$:
 - (a) Genera un campione bootstrap dei dati di training
 - (b) Genera un albero random forest per i dati bootstrapped ripetendo ricorsivamente i seguenti passi
 - i. Seleziona casualmente m variabili dalle p disponibili
 - ii. Tra queste m , scegli la variabile che restituisce lo split migliore
 - iii. Dividi il nodo in due nodi “figli”
 4. Usa il criterio di stop scelto per determinare quando un albero è completo
 5. Restituisci l'ensemble di alberi
-

numerosità pari a quella del training set, effettuato proprio su quest'ultimo. Inoltre ognuno di questi dataset ha un corrispettivo *out-of-bag (oob) set*, composto dalle unità del training set non rientrate nel bootstraped set e per questo utilizzabili al pari del test set.

A questo punto l'algoritmo scelto viene allenato su ognuno di questi bootstrapped dataset e, infine, viene effettuata la media (nel caso della regressione) di tutti i risultati ottenuti. Questa azione è detta “bagging” e le foreste casuali sono ensemble di alberi di decisione allenati tramite bagging.

A seguito del bagging su alberi decisionali si possono riscontrare correlazioni tra i vari alberi, limitando così l'effetto della riduzione della varianza. Per evitare ciò, l'algoritmo del random forest introduce un ulteriore fattore di casualità che porta a una decorrelazione degli alberi. Nella costruzione dei singoli alberi, la variabile di split a ogni nodo non viene scelta considerando tutte le p features disponibili, come negli alberi di decisione, ma viene scelta da un campione casuale di $m < p$ variabili. Solitamente viene posto $m = \sqrt{p}$ se il problema è di classificazione, $m = \frac{p}{3}$ se di regressione. Se $m = p$ si ha semplicemente un algoritmo di bagging.

L'algoritmo di base del random forest per un problema di regressione o classificazione è mostrato in Hastie et al. (2017) e qui riportato come Algoritmo 1.

Iperparametri Random Forest Con il termine *iperparametri* si intendono tutte quelle variabili i cui valori, all'interno dell'algoritmo, possono essere cambiati per ottenere performance migliori rispetto al modello “base”.

I valori finali vengono solitamente scelti applicando delle specifiche strategie di tuning che permettono di iterare più volte l'algoritmo scelto cambiando ogni volta uno o più di questi iperparametri. È immediato intuire come questo processo possa estendere i tempi di elaborazione dell'output ma in molti casi mettere in

pratica questa operazione può rivelarsi un'ottima strategia per ottenere performance migliori.

Ogni iperparametro ha, inoltre, un proprio impatto sui risultati quindi è sempre bene studiare la documentazione ufficiale prima di procedere con qualsiasi strategia di tuning.

Seguendo il lavoro di Probst et al. (2019), nel caso del random forest i principali iperparametri sono:

- Il numero di alberi nella foresta (n_tree): buona norma è iniziare usando $n_tree = 10 \times p$;
- Il numero di variabili da considerare a ogni split: di default si ha $m = \frac{p}{3}$ nel caso di regressione;
- La complessità degli alberi: data da “*node size*” (di default pari a 5 per la regressione) o “*max depth*” (Segal, 2004; Díaz-Uriarte & Alvarez de Andrés, 2006; Goldstein et al., 2011);
- Lo schema di campionamento: di default viene utilizzato il bootstrapping.

In questo caso l'iperparametro con maggior impatto è m , parametro che quindi è sempre meglio inserire nella strategia di tuning. Gli altri hanno un'influenza minore sull'output finale ma non è conveniente escluderli dal tuning.

2.2.3 XGBoost

L'*Extreme Gradient Boosting* (abbreviato *XGBoost*) è una ottimizzazione del modello *Gradient Boosting*, derivante, a sua volta, dal *Boosting*.

In 2.2.2 si è visto come il bagging implica la creazione di nuovi dataset di training tramite la tecnica bootstrap, l'evoluzione di un albero decisionale per ogni nuovo dataset e, infine, la combinazione di tutti gli alberi per restituire un unico output. Si ha quindi una elaborazione degli alberi in parallelo.

Il *boosting* segue un procedimento simile ma, al contrario, gli alberi sono addestrati in modo sequenziale, ossia ogni albero viene elaborato utilizzando gli errori degli alberi precedenti, migliorando quindi le prestazioni del modello mano a mano che vengono costruiti nuovi alberi. A ogni iterazione viene data quindi maggiore importanza ai dati su cui il modello performa peggio l'output finale è dato da una media ponderata dei risultati dei singoli alberi elaborati.

Il metodo boosting è stato utilizzato come punto di partenza per sviluppare gli algoritmi *Adaptive Boosting* (chiamato *AdaBoost*) (Freund & Schapire, 1997) e *Gradient Boosting* (Friedman, 2001).

Quest'ultimo migliora il boosting con l'introduzione della *discesa del gradiente* come algoritmo di ottimizzazione per trovare la soluzione ottima. Misura iterativamente il gradiente della funzione di costo in un punto dato da specifici parametri e modifica tali valori cercando di minimizzare la suddetta funzione. In questo contesto assume molta importanza il *learning rate*, parametro che controlla la grandezza degli spostamenti sulla curva della funzione di perdita. Un valore troppo basso porta all'ottenimento di un modello più accurato a fronte di un incremento di iterazioni e

quindi a una crescita dei tempi di elaborazione dell'output. Dall'altra parte, un valore troppo alto può comportare il salto dell'effettivo valore minimo della funzione, potenzialmente restituendone un valore più alto di quello iniziale.

Il modello gradient boosting viene migliorato e ottimizzato nel 2016 con l'introduzione dell'XGBoost grazie a Chen & Guestrin.

Ciò che principalmente lo contraddistingue dall'algoritmo di origine sono gli iperparametri, descritti brevemente nella sezione successiva.

Iperparametri XGBoost Come riportato anche nella documentazione ufficiale, XGBoost riprende gli iperparametri del boosting e degli algoritmi ad albero ma con alcuni vantaggi:

- Regolarizzazione: sono implementati tre parametri che aiutano a ridurre la complessità dell'algoritmo e a evitare l'overfitting
 - Gamma: definito come moltiplicatore lagrangiano. Controlla la complessità di un singolo albero e indica la riduzione minima di perdita richiesta per procedere con la partizione del nodo. Varia tra 0 e ∞ ;
 - Alpha: limita l'influenza che possono avere le singole foglie di un albero sul risultato finale. Varia tra 0 e ∞ ;
 - Lambda: stessa funzione e stessi valori di *alpha*.
- Arresto anticipato: l'algoritmo può essere interrotto anticipatamente se i nuovi alberi non offrono miglioramenti.
- Elaborazione in parallelo: il modello implementa una procedura per parallelizzare, e quindi velocizzare, il processo di calcolo sfruttando la GPU o Apache Spark
- Funzione di perdita: permette di utilizzare funzioni di costo customizzate
- Multilinguaggio: è stato sviluppato per vari ambienti quali R, Python, Julia, Java e C++

Come avviene per gli iperparametri del random forest, anche in questo caso è consigliato attuare una strategia di tuning per trovare i valori che restituiscano il miglior risultato possibile.

2.2.4 Interpretabilità del Modello

Una volta creato il modello, effettuato l'eventuale tuning e ottenuto un output, fondamentale diventa il saper leggere e descrivere come l'algoritmo si è comportato, riuscendo così ad avere maggiore chiarezza anche sui risultati ottenuti.

Di questo si occupano tutte quelle analisi rivolte all'interpretazione dei modelli, cercando di capire più nel dettaglio cosa, come e perché è stato restituito quel preciso output.

Per evitare incoerenza con gli obiettivi di questo testo, nel seguito sono riportate solo quelle analisi che sono state utilizzate nello studio su cui si basa questo lavoro.

Feature Importance

Con il termine *feature importance* si intende quel tipo di analisi che ha come obiettivo la quantificazione dell'impatto delle singole variabili sul risultato finale.

Segue un distinguo tra i modelli visti nella sezione precedente

Random Forest Nel caso del random forest viene ripreso il calcolo effettuato in un modello di bagging e implementato con un'analisi permutativa.

Per il bagging la procedura prevede, per ogni albero, la somma, per ogni variabile utilizzata nei partizionamenti, della riduzione della funzione di perdita associata a quella specifica variabile. Questo valore viene poi aggregato, per ogni variabile, tra tutti gli alberi. Le caratteristiche con la maggiore riduzione media, in riferimento alla funzione di perdita, sono considerate le più importanti.

Nel random forest viene ripresa la procedura del bagging implementandolo facendo passare, per ogni albero, il campione out-of-bag lungo tutto l'albero registrando l'accuratezza della previsione. Dopodiché vengono permutati casualmente i valori di ogni variabile (una alla volta) e viene nuovamente calcolata l'accuratezza con il campione oob. La differenza tra i due valori di accuratezza restituisce l'importanza della variabile i cui valori sono stati permutati. Le features su cui viene registrata la maggior differenza sono quelle più importanti.

XGBoost Essendoci alla base una struttura ad albero, il calcolo della feature importance per XGBoost segue quanto detto nel caso del random forest. In questo caso si hanno però a disposizione tre tipologie di misura:

- Gain: equivalentemente al random forest, indica il miglioramento dell'accuracy del modello apportato da una variabile;
- Copertura: quantifica il numero relativo di osservazioni influenzate da ciascuna caratteristica
- Frequenza: rappresenta il numero relativo di volte che ciascuna variabile si presenta negli alberi del modello

Dipendenza Parziale

Ulteriore aiuto nella lettura di un modello viene dalla dipendenza parziale.

Questa misura, assieme al grafico associato, chiamato *partial dependence plot (PDP)*, aiuta a capire l'effetto marginale di una variabile sul risultato previsionale medio di un algoritmo, consentendo di capire se la relazione tra l'output e la variabile considerata sia lineare o più complesso. Ciò è possibile in quanto l'output del modello viene marginalizzato rispetto la distribuzione delle variabili che non interessano, mantenendo quindi la relazione tra la variabile di interesse e il risultato, restituendo così un'idea di come varia l'output considerando un effetto medio delle altre variabili.

La funzione di dipendenza parziale viene descritta da Friedman (2001), e spiegata da Molnar (2020), come:

$$f_S(x_S) = E_{X_C} [f(x_S, X_C)] = \frac{1}{n} \sum_{i=1}^n f(x_S, x_C^{(i)}).$$

x_S sono le variabili di interesse, quelle per cui si vogliono conoscere gli effetti sulla previsione. S solitamente contiene al massimo, per motivi grafici, due variabili. X_C sono tutte le altre variabili utilizzate nel modello (e nella funzione considerate come aleatorie).

Per capirne come viene implementata questa funzione, sono riportati l'algoritmo base del calcolo della dipendenza parziale per un singolo predittore (Algoritmo 2) e un'immagine che rende chiaro il meccanismo (Figura 2.5, in rosso la variabile di interesse).

La dipendenza parziale ha vantaggi e svantaggi.

Vantaggi

- Facile interpretazione: la funzione di dipendenza parziale rappresenta la previsione media se si forzassero tutti i dati ad assumere quel determinato valore per la variabile di interesse.
- L'interpretazione è di tipo causale, si modifica una variabile e si misurano le variazioni nella previsione. Importante però è sottolineare come la relazione sia causale per il modello e non necessariamente anche per il mondo reale. È possibile stabilire delle ipotesi per raggiungere quest'ultimo risultato, come spiegato da Zhao & Hastie (2021), ma le metodologie richiedono uno studio specifico che va oltre gli scopi di questo lavoro.

Svantaggi

- Se la variabile per cui viene calcolata la dipendenza parziale è fortemente correlata con le altre variabili, possono esserci problemi di rappresentazione. Apley & Zhu (2020) suggeriscono, come possibile soluzione, di utilizzare l'*Accumulated Local Effect plot* (ALE plot) che si basa sulla distribuzione condizionale invece che sulla marginale.
- Vengono nascosti eventuali effetti eterogenei in quanto sono rappresentati gli effetti marginali medi. Goldstein et al. (2015) suggeriscono di utilizzare le *Individual Conditional Expectation curves* (ICE).

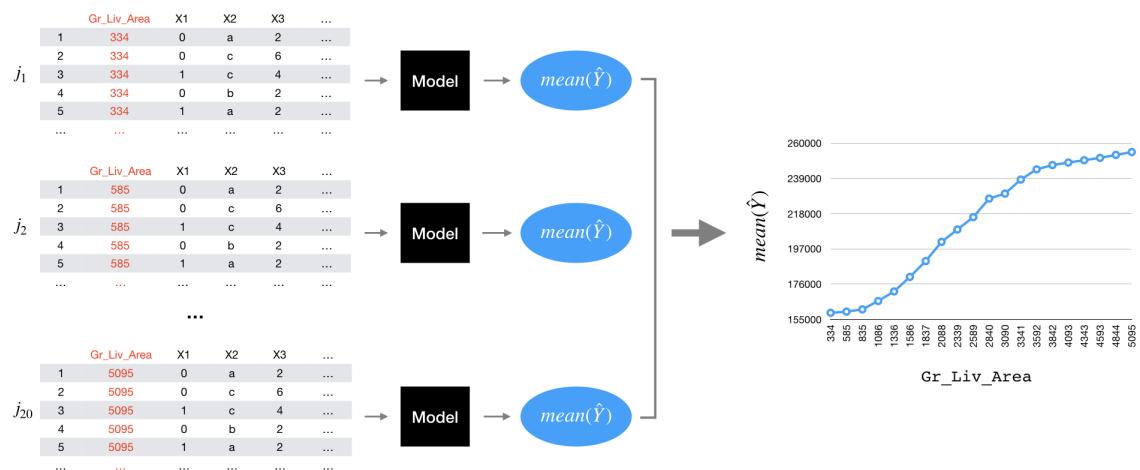
2.3 Apprendimento non Supervisionato

«If we fit a predictive model using a supervised learning technique, then it is possible to *check our work* by seeing how well our model predicts the response Y on observations not used in fitting the model. However, in unsupervised learning, there is no way to check our work because we don't know the true answer – the problem is unsupervised»².

²James et al., *An Introduction to Statistical Learning*, Springer, 2021², p. 498

Algoritmo 2 Dipendenza ParzialeDato un predittore x

1. Costruire una griglia di j valori uniformemente distanziati attraverso la distribuzione di $x : \{x_1, x_2, \dots, x_j\}$
2. Per i in $\{1, \dots, j\}$:
 - (a) Crea una copia del training set e sostituisci i valori di x con la costante x_i
 - (b) Applica il modello scelto
 - (c) Fai la media delle previsioni ottenute per ogni unità
3. Plotta le previsioni medie ottenute rispetto i valori x_1, \dots, x_j

**Figura 2.5.** PDP ([80])

Questa frase mette in risalto la differenza esistente tra le due tipologie di metodologie di analisi.

Nell'apprendimento supervisionato si hanno m variabili, osservate su n unità, e una varabile risposta Y , osservata anch'essa sulle n unità, e l'obiettivo è ottenere una previsione di Y sfruttando le X_1, \dots, X_m .

Nell'apprendimento non supervisionato si hanno le variabili X_1, \dots, X_m a cui però non è associata alcuna Y su cui cercare di effettuare previsioni. L'obiettivo infatti è cercare informazioni su eventuali relazioni esistenti tra le m variabili.

Per la mancanza di una Y di riferimento, questa tipologia di dati è chiamata *unlabeled*.

Questa tipologia di analisi, a differenza di quella supervisionata, ha molteplici utilizzi che si possono riassumere in tre categorie, elencate di seguito e per cui sono indicati gli algoritmi più utilizzati:

- Clustering: l'obiettivo è raggruppare unità con caratteristiche simili
 - Clustering Partizionale (*Partitioning Clustering*)
 - Clustering Gerarchico (*Hierarchical Cluster Analysis, HCA*)
 - Clustering Spettrale (*Spectral Clustering*)
 - Modelli a Mistura Gaussiana (*Gaussian Mixture Model*)
 - DBSCAN;
- Anomaly detection: l'obiettivo è scovare unità che si discostano dalle altre (cfr. Sez. 2.1.2)
 - One-class SVM
 - Isolation Forest
 - Local Outlier Factor;
- Riduzione dimensionalità
 - Analisi delle Componenti Principali (*Principal Component Analysis, PCA*)
 - Kernel PCA
 - Autoencoders.

Nelle analisi effettuate in questo studio si è scelto di introdurre solamente analisi di cluster per rilevare eventuali pattern interni ai dati, e nella prossima sezione sarà approfondita esclusivamente questa tipologia di algoritmi, con maggiore focus riguardo l'algoritmo che si è deciso di utilizzare.

2.3.1 Clustering

Non c'è una definizione universale di “clustering”. Dipende molto dal contesto e dal tipo di algoritmo che si adotta. La cluster analysis è infatti una tecnica utilizzata in diversi settori, come segmentazione della clientela per sistemi di raccomandazione, ricerca per immagini, segmentazione di immagini per object detection ecc.

Tutti i possibili obiettivi, però, riguardano il raggruppare o segmentare una collezione di oggetti in sottoinsiemi, detti “cluster”, in modo tale che quelli all'interno

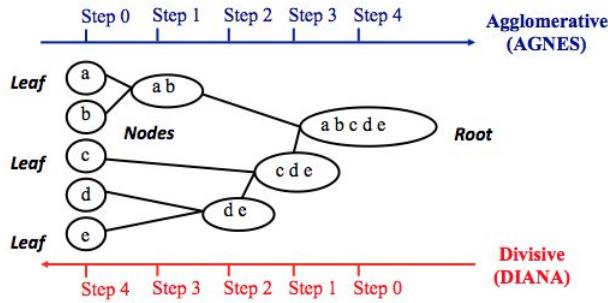


Figura 2.6. Clustering agglomerativo vs divisivo ([81])

di ogni cluster sono più connessi tra loro rispetto agli oggetti che si trovano in altri cluster.

Questa “connessione” tra unità è data dal grado di *dissimilità*, una funzione che associa a ciascuna coppia di unità un valore numerico, rappresentante della diversità tra le due unità.

Fondamentale diventa quindi la scelta della misura di dissimilità, affrontata in 2.3.1

Per quanto concerne gli algoritmi utilizzati nel clustering, le categorie di clustering più diffuse sono due: partizionale e gerarchico.

Il clustering partizionale richiede all’analista di specificare il numero K di cluster da realizzare. K è quindi un input del modello.

Gli algoritmi più utilizzati in questo contesto sono:

- *K-Medie*: ogni cluster è rappresentato dal centro, detto *centroide*, dei punti appartenenti al cluster. Solitamente il centroide corrisponde alla media (MacQueen, 1967);
- *K-Medoide PAM*: ogni cluster è rappresentato da una unità interna al cluster, detta *medoide* (Kaufman & Rousseeuw, 1990);
- *CLARA*: estensione del K-Medoide riaddattato per dataset con un gran numero di dati;
- *DBSCAN*: a differenza dei precedenti, crea cluster basandosi sulla densità dei punti in una determinata area (cfr. Sez. 2.1.2).

Il clustering gerarchico, invece, non necessita di avere in input il numero di cluster in quanto questo valore sarà scelto a posteriori in base a determinati criteri.

Come si può intuire dal nome, questa tipologia di algoritmi produce una rappresentazione gerarchica in cui il cluster presente a ogni livello è ottenuto unendo più cluster del livello inferiore. Al livello più basso ogni cluster contiene una singola osservazione, al livello più alto è presente un solo cluster che contiene tutte le unità.

In questo caso esistono due metodologie:

- *Clustering Aggregativo* (o *agglomerativo*): inizialmente ogni osservazione rappresenta un cluster a sé stante. Cluster simili vengono successivamente uniti fino a ottenere un solo cluster contenente tutte le unità.

- *Clustering Divisivo*: iniziando includendo tutte le unità in un solo cluster, si conclude con osservazioni che costituiscono singoli cluster.

Il risultato di quest'ultima tipologia di clustering è una rappresentazione ad albero detta *dendrogramma*, utilizzato, assieme ad altre informazioni, per decidere il livello a cui tagliare l'albero e ottenere quindi il numero di cluster finali.

Dissimilità

Date n unità, la dissimilità d tra due unità $x_i, x_k (i, k = 1..n)$ è una funzione che associa alla coppia (x_i, x_k) un valore numerico che quantifica la loro diversità, o dissimilità, se è:

- non negativa: $d_{ik} \geq 0$;
- nulla: $d_{ii} = 0$;
- simmetrica: $d_{ik} = d_{ki}$.

Questi valori sono poi riportati in una matrice quadrata, detta *matrice di dissimilità* D :

$$D = \begin{bmatrix} 0 & d_{12} & \cdots & d_{1i} & \cdots & d_{1k} & \cdots & d_{1n} \\ d_{21} & 0 & \cdots & d_{2i} & \cdots & d_{2k} & \cdots & d_{2n} \\ \vdots & \vdots & 0 & \vdots & \cdots & \vdots & \cdots & \vdots \\ d_{i1} & d_{i2} & \cdots & 0 & \cdots & d_{ik} & \cdots & d_{in} \\ \vdots & \vdots & \cdots & \vdots & 0 & \vdots & \cdots & \vdots \\ d_{k1} & d_{k2} & \cdots & d_{ki} & \cdots & 0 & \cdots & d_{kn} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots & 0 & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{ni} & \cdots & d_{nk} & \cdots & 0 \end{bmatrix}$$

Se inoltre viene soddisfatta la diseguaglianza triangolare $d_{ik} \leq d_{il} + d_{kl}$ per $i, l, k = 1..n$, la diversità è detta *distanza*. Raramente però si verifica questa condizione.

Riprendendo la notazione di Hastie et al. (2017), lavorando in uno spazio p -dimensionale e considerando la j -esima variabile ($j = 1..p$), la dissimilità è definita come $d_j(x_{ij}, x_{kj})$. Da questa si ottiene la dissimilità tra le due unità su tutte le p variabili, $D(x_i, x_k) = \sum_{j=1}^p d_j(x_{ij}, x_{kj})$.

Nello specifico la dissimilità viene scelta in base alla tipologia delle variabili, e spesso si preferisce assegnare un peso diverso ad ogni variabile. Si ha pertanto:

- Variabili qualitative.

In caso di variabili binarie, le modalità delle unità i e k sono disposte nelle matrici C_i e C_k , ciascuna di dimensione $p \times 2$.

Viene quindi costruita la matrice $C'_i C_l = \begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix}$ in cui c_{uv} rappresenta

la frequenza delle variabili che presentano, per la coppia di unità (i, k) , le modalità (u, v) con $u, v = \{0, 1\}$. Da questa matrice ottengo le informazioni per la misura delle dissimilità, quali:

- Jaccard-Needham: $\frac{c_{10}+c_{01}}{c_{11}+c_{10}+c_{01}}$;
- Yule: $\frac{c_{10}\times c_{01}}{c_{11}\times c_{00}+c_{10}\times c_{01}}$;
- Distanza Euclidea: $\sqrt{c_{10} + c_{01}}$;
- Varianza: $\frac{c_{10}+c_{01}}{4\times p}$.

- Variabili quantitative.

Data una matrice dei pesi W e un parametro r , le misure più frequentemente utilizzate sono:

- Distanza di Minkowski di ordine r : $\sqrt[r]{\sum_{j=1}^p |x_{ij} - x_{kj}|^r \times w_j}$;
- Distanza della città a blocchi: $\sum_{j=1}^p |x_{ij} - x_{kj}| \times w_j$;
- Distanza Euclidea: $\sqrt{\sum_{j=1}^p |x_{ij} - x_{kj}|^2 \times w_j}$;
- Distanza di Camberra: $\frac{|x_{ij} - x_{kj}|}{(|x_{ij}| + |x_{kj}|)}$.

- Variabili miste.

In questo caso si hanno due possibili approcci:

- Convertire tutte le variabili nella tipologia più frequente;
- Media ponderata delle dissimilarità delle p variabili: $d_{ik} = \frac{\sum_{j=1}^p w_{ikj} \times d_{ikj}}{\sum_{j=1}^p w_{ikj}}$.

A prescindere dalla misura di dissimilarità utilizzata, è sempre consigliato ricorrere a una standardizzazione dei dati prima di calcolare la matrice D . Nel caso delle variabili quantitative, W può essere utilizzata come matrice dei pesi o di standardizzazione.

Clustering Aggregativo

Come accennato precedentemente, il clustering aggregativo lavora con una strategia *bottom-up*, partendo considerando ogni unità come singolo cluster e aggregando, con il procedere dei passi dell'algoritmo, cluster che sono “simili”. Questa procedura si reitera fino a ottenere un unico cluster contenente tutte le osservazioni.

Riprendendo il formalismo di Kassambara (2017), i passi per ottenere l'output di un'operazione di clustering di questo tipo prevedono:

1. Preparazione dei dati (standardizzazione)
2. Calcolo della dissimilarità tra i dati e ottenimento della matrice D (cfr. Sez.2.3.1)
3. Fusione dei cluster tramite calcolo della loro differenza (funzione legame o *linkage function*)
4. Scelta del numero di cluster finali tramite taglio del dendrogramma
5. Verifica della correttezza dei cluster.

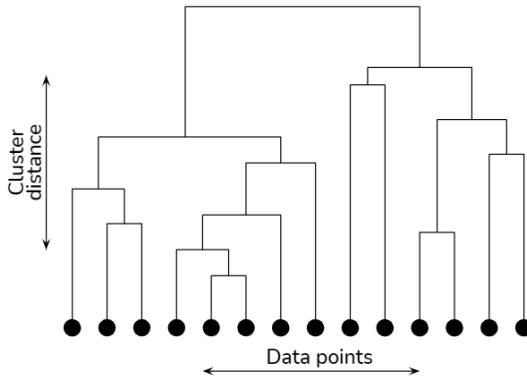


Figura 2.7. Dendrogramma ([82])

Funzione Legame Con il termine *linkage function* si intendono tutti quei metodi aggregativi che permettono di calcolare la distanza tra i cluster in esame e unirli in base alle caratteristiche del metodo scelto.

Posti X e Y i due cluster da esaminare, i metodi di legame più diffusi sono:

- Legame singolo (*single linkage*): la distanza tra X e Y è definita dalla minima distanza calcolata tra i loro elementi. $D_{X,Y} = \min_{x \in X, y \in Y} d(x,y)$.
- Legame completo (*complete linkage*): la distanza tra X e Y è definita dalla massima distanza calcolata tra i loro elementi. $D_{X,Y} = \max_{x \in X, y \in Y} d(x,y)$.
- Legame medio (*average linkage*): la distanza tra X e Y è definita dalla distanza media calcolata tra i loro elementi. $D_{X,Y} = \frac{1}{|X||Y|} \sum_{x \in X} \sum_{y \in Y} d(x,y)$.
- Metodo di Ward (*Ward's method*): mira a minimizzare la varianza intra-cluster unendo a ogni passo i cluster con la minima distanza inter-cluster. $D_{X,Y} = \frac{|X| \times |Y|}{|X| + |Y|} \|\bar{X} - \bar{Y}\|^2$, con \bar{X} e \bar{Y} valore centrale del cluster.

Dendrogramma Il dendrogramma ha la funzione di rappresentare graficamente le unità e i legami tra di esse, restituendo la forma di un albero in cui le foglie sono le singole unità e i rami costituiscono i collegamenti tra le unità (Figura 2.7).

Muovendosi infatti dal basso verso l'alto, le osservazioni simili (secondo la definizione di dissimilarità vista sopra) sono collegate tramite rami, a loro volta uniti ad altri rami mano a mano che si sale. L'altezza a cui si uniscono i rami indica la distanza tra i due cluster e quindi più è alto il valore di questa altezza, minore è la somiglianza tra i cluster.

Per ottenere la divisione in cluster, si taglia il dendrogramma a una determinata altezza, spesso data dal cosiddetto *metodo del gomito* (o *elbow method*). Esistono altre metodologie per determinare il numero di cluster da considerare, anche più precise del “gomito”, ma si rimanda a testi specifici per tali approfondimenti.

Utilizzare il metodo del gomito prevede la creazione di un grafico (detto *scree-plot*) avente sull'asse delle ascisse il numero di cluster e in ordinata le distanze tra i cluster. È quindi disegnata una spezzata che congiunge i punti di intersezione tra i

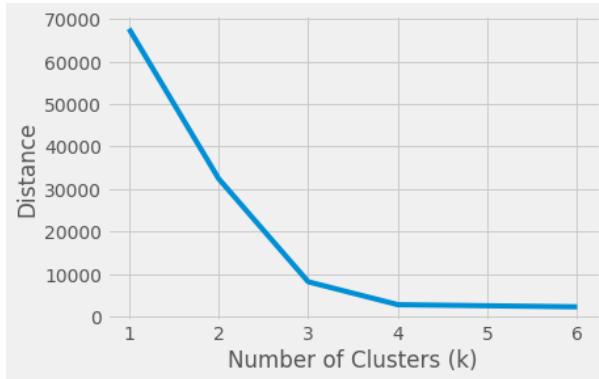


Figura 2.8. Metodo del gomito ([83])

due assi, in accordo con quanto rappresentato dal dendrogramma. Come numero di cluster viene scelto quel valore che crea nel grafico un gomito, cercando il miglior compromesso tra minor numero di cluster ma massima distanza tra di essi (in Figura 2.8 si potrebbe scegliere $k = 3$ o $k = 4$).

Silhouette Per verificare la correttezza della divisione in cluster esistono diverse metodologie, che però in questo testo non trovano spazio per essere approfondite tutte.

La misura di interesse per questo lavoro è il *coefficiente di silhouette*.

L’analisi della silhouette misura quanto un’unità è stata ben clusterizzata e la sua rappresentazione grafica restituisce un’idea di quanto l’unità in un cluster sia simile a unità del cluster vicino.

Riprendendo, anche in questo caso, la notazione di Kassambara (2017), il valore della silhouette è calcolato come:

1. Per ogni osservazione i , calcolare la dissimilarità a_i tra i e tutti gli altri punti del cluster a cui appartiene l’unità;
2. Per tutti i cluster C a cui non appartiene i , calcola la dissimilarità media $d(i, C)$ tra i e tutte le osservazioni di C . La più piccola è definita come $b_i = \min_C d(i, C)$. b_i rappresenta la dissimilarità tra i e il cluster più vicino;
3. La silhouette per l’unità i è quindi definita come $S_i = \frac{b_i - a_i}{\max(a_i, b_i)}$.

Osservando i valori S_i così ottenuti, si può capire se le osservazioni sono ben clusterizzate. Infatti

- Se S_i tende a 1 si può concludere che l’unità i appartiene correttamente a quel cluster.
- Se $S_i \approx 0$, i giace tra due cluster.
- Se $S_i < 0$, probabilmente i è rientrato nel cluster sbagliato ed è consigliato spostarlo nel cluster vicino.

2.4 Apprendimento per Rinforzo

L'apprendimento per rinforzo, conosciuto meglio con il termine *reinforcement learning* (RL), costituisce una branca piuttosto recente del machine learning.

Nonostante i primi lavori siano quelli di Sutton & Barto (1998), solo negli ultimi dieci anni si è vista una crescita esponenziale di lavori basati su questa tipologia di modelli.

In questo caso il sistema di apprendimento è chiamato *agent* ed è capace di osservare l'ambiente, selezionare ed eseguire azioni per ottenere in cambio ricompense o penalità. All'agent non viene però detto a priori quali azioni deve intraprendere ma tramite un sistema di “trial and error” impara autonomamente la strategia migliore, detta *policy*, per ottenere la maggior ricompensa possibile. La *policy*, quindi, determina l'azione che l'*agent* deve eseguire quando si trova in una determinata situazione.

L'apprendimento per rinforzo è quindi diverso dall'apprendimento supervisionato in quanto in quest'ultimo il sistema di apprendimento si basa su dati etichettati e l'etichetta rappresenta esattamente l'azione corretta che il sistema dovrebbe fare in una situazione simile a quella descritta dai predittori. Nell'apprendimento per rinforzo non è possibile avere una situazione simile in quanto difficilmente si riescono a ottenere dati etichettati per tutte le situazioni in cui l'*agent* dovrebbe agire, e deve quindi essere in grado di imparare dall'esperienza.

Risulta diverso anche dall'apprendimento non supervisionato, e in questo caso la diversità risiede negli obiettivi. L'apprendimento non supervisionato va alla ricerca di strutture trai dati non etichettati, come somiglianze e differenze. Il reinforcement learning cerca invece di trovare quella serie di azioni che permettano di massimizzare la ricompensa totale.

In questo contesto assumono un ruolo centrale alcuni termini che vanno a descrivere l'ambiente in cui lavora un tale algoritmo:

- Ambiente: l'ambiente fisico in cui opera l'*agent*;
- Stato: situazione corrente dell'*agent*;
- Ricompensa: feedback ricevuto dall'ambiente;
- Policy: metodo che mappa lo stato dell'*agent* rispetto le azioni;
- Valore: ricompensa futura che l'*agent* riceverebbe compiendo un'azione in un determinato stato.

Per come si configura, risultano chiare le potenzialità di applicabilità di questo algoritmo in molteplici settori, quali, ad esempio:

- Auto a guida autonoma (Balaji et al., 2019; Kiran et al., 2021);
- Natural Language Processing (Grissom II et al., 2014; Li et al., 2016; Paulus et al., 2017);
- Sanità (Yu et al., 2019);
- Sistemi di produzione (Gauci et al., 2018);

- Sistemi di raccomandazione (Zheng et al., 2018);
- Gaming (Silver et al., 2017);
- Marketing (Jin et al., 2018).

Per le finalità di questo lavoro non si ritiene opportuno approfondire maggiormente questo argomento e si rimanda a *Reinforcement Learning: An Introduction* di Sutton & Barto, considerato uno dei migliori libri per iniziare a studiare questa tipologia di algoritmi.

2.5 Descrizione Dataset

Il 19 gennaio 2014 ci fu una rottura nell’argine ovest del fiume Secchia nei pressi della frazione San Matteo, zona settentrionale della provincia di Modena.

Secondo gli studi di Orlandini et al. (2015), in meno di 30 ore, circa 37×10^6 m³ d’acqua invasero i comuni limitrofi, inondando un’area complessiva di 52 m² (Figura 2.9). Le zone colpite rimasero sott’acqua per più di 48 ore, subendo un danno complessivo pari a circa 500 Milioni di Euro.

Tra i comuni più colpiti e di cui si è in possesso di un volume più consistente di dati vi sono Bastiglia e Bomporto. Ed è proprio su questi che si sono concentrati questo e i passati studi.

Quest’area è situata lungo la sponda destra del fiume e si estende per circa 112 km². Da un punto di vista geomorfologico, è prevalentemente pianeggiante e i principali rilievi sono costituiti da strade o ferrovie e argini fluviali secondari. Orientata verso nord-est, la zona subisce una leggera escursione altimetrica, passando dai circa 30 m s.l.m. nella parte sud occidentale ai circa 18 m s.l.m. in direzione nord-est. L’area di interesse è delimitata a ovest dall’argine destro del fiume Secchia, a est dall’argine sinistro del fiume Panaro, a nord e a sud da strade e canali secondari che hanno evitato ulteriori allagamenti nella zona settentrionale.

I primi segni di rottura sono stati rilevati intorno alle 06:30 e, sempre secondo le analisi di Orlandini, la possibile causa è il collasso di un sistema di tane di nutrie. A seguito delle ingenti piogge dei giorni precedenti, questa parte di argine è scesa di 1 metro sotto il livello del fiume ed entro le 15:00 il fiume ha raggiunto l’altezza dei terreni oltre l’argine.

Grazie a testimonianze, video e studi specifici è stato possibile risalire alla dinamica dell’evento potendo realizzare così modelli idrodinamici piuttosto accurati. Aggregando i dati provenienti da questi modelli e da indagini delle autorità competenti si è riusciti a collezionare dati importanti per diverse tipologie di analisi, tra cui quella presentata in questo lavoro.

2.5.1 I Dati dalle Schede di Danno

A seguito dell’evento, le autorità della Regione Emilia-Romagna, della Provincia di Modena e dei comuni interessati hanno avviato una raccolta dati per ottenere informazioni, a fini di risarcimento, sui danni causati dall’evento. A cittadini e proprietari immobiliari sono state somministrate delle “schede di danno” (Figura

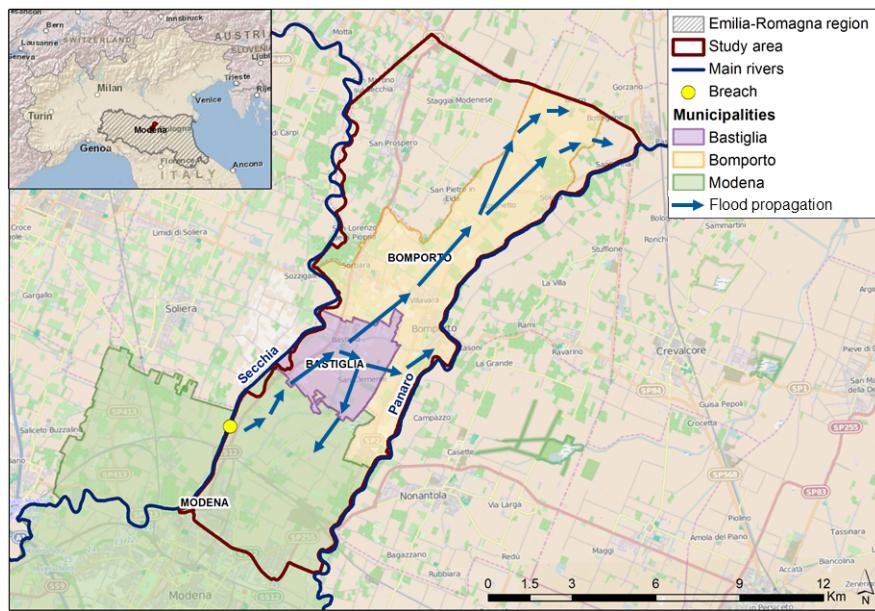


Figura 2.9. Zona inondata ([13])

2.10) relative a danni a cose pubbliche, beni immobili, mobili e registrati, danni ad attività economiche e produttive.

A seguito della scarsità di alcune informazioni e per gli obiettivi del presente studio, sono stati presi in considerazione esclusivamente danni a beni immobili privati, per un totale di 1.918 osservazioni, 1.349 nel Comune di Bastiglia, 569 in quello di Bomporto. È stato però necessario eliminarne alcune a causa della mancanza di dati per variabili fondamentali nell’analisi, come la posizione geografica, o perché considerate come valori anomali. Questo aspetto verrà approfonditò nella sezione apposita (cfr. Sez. 3.1).

Dalle schede sono stati estrapolati dati inerenti a:

- Danni in termini economici riguardo l’immobile
- Tipologia della struttura dell’immobile (in muratura, cemento armato, cemento e muratura, laterocemento, a facciavista)
- Superficie dell’immobile

Queste informazioni sono state ampliate con quelle provenienti dai registri OMI (Osservatorio del Mercato Immobiliare) ottenendo così il valore economico di ciascun immobile, sia in termini di costruzione che di compravendita. Utilizzando quest’ultimo è stato possibile calcolare il danno relativo per ogni edificio, considerandolo come percentuale del danno subito rispetto al valore totale dell’immobile. Riprendo le parole di Carisi et al. (2018), «these economic values do not consider a possible fall in price due to catastrophic events. Also, we are aware that reconstruction costs seem to be more suitable for this kind of analyses, but they are not freely available in Italy or homogeneous at a national level, different from OMI values». A supporto dell’utilizzo del danno relativo sono inoltre presenti lavori precedenti (Ar-

COMUNE DI BASTIGLIA

n. progressivo scheda B: [REDACTED]

SCHEDA B

**Ricognizione del fabbisogno per il
ripristino del patrimonio edilizio privato, dei beni mobili e dei
beni mobili registrati**

REGIONE EMILIA ROMAGNA**EVENTI ALLUVIONE DEL 19/01/2014****SEGNALAZIONE E QUANTIFICAZIONE DEL DANNO**

(Autocertificazione ai sensi del D.P.R. 445/2000)

COMUNE DI BASTIGLIA PROVINCIA MO

Il sottoscritto [REDACTED]

nato/a a [REDACTED] il [REDACTED]

residente a BASTIGLIA CAP 41030 Indirizzo [REDACTED]

Tel. _____; Cell. [REDACTED]; Fax. _____

consapevole delle conseguenze penali previste dall'art.76 del D.P.R. 445/2000 per le falsità in atti e le dichiarazioni mendaci

**DICHIARA
SOTTO LA PROPRIA RESPONSABILITA'**

1) che l'immobile è ubicato in

via / viale / piazza [REDACTED] n. civico: [REDACTED];

località: _____ BASTIGLIA _____ CAP 41030 _____

**L'immobile è
X di proprietà in comproprietà
(nome del comproprietario: _____)**

 altro diritto reale di godimento (specificare: _____)**COMUNE DI BASTIGLIA****Figura 2.10. Esempio scheda di danno**

righi et al., 2013; Domeneghetti et al., 2015) secondo i quali questi valori economici risultano ancora informativi per una stima ex ante del danno.

2.5.2 I Dati da Modello SWAM

Al fine di implementare i dati raccolti con una stima dell'altezza dell'acqua nei diversi punti della mappa, è stato utilizzato il metodo SWAM (*Surface Water Analysis Method*), realizzato in ambiente GIS (*Geographic Information System*) e sviluppato da Pastormerlo (2016).

Tramite parametrizzazione degli input (si veda Pastormerlo, 2016), si ottiene in output il valore dell'altezza dell'acqua in ogni punto dell'area considerata.

Questa metodologia richiede in input le informazioni morfologiche della zona (ottenibili tramite un *Digital Terrain Model*, DTM) e uno shapefile della zona alluvionata (reperibile tramite Autorità di Bacino). Inoltre prevede due assunti: nei limiti estremi dell'alluvione l'altezza dell'acqua è considerata nulla e in punti simmetrici rispetto l'alveo fluviale se ne riscontra una quota comparabile.

In un contesto urbano e reale queste due condizioni non sempre sono verificate e pertanto si può ipotizzare una possibile distorsione delle stime ottenute rispetto ai livelli osservati durante l'evento. Inoltre il DTM restituisce una superficie morfologica priva di ostacoli, siano essi naturali o antropici. Ma è possibile che interventi umani abbiano comportato modifiche nella morfologia del terreno. Per l'estensione dell'area considerata queste possibili distorsioni sono però considerate poco influenti.

Se sono disponibili dati puntuali osservati per questa variabile, è possibile integrarli nel modello per migliorarne la precisione.

2.5.3 I Dati da Ambiente GIS

I 1.918 dati raccolti sono stati caricati in ambiente GIS tramite il software QGIS (<https://qgis.org/>), applicativo utile per la creazione, gestione e analisi di mappa. All'interno di questo ambiente, i dati sono stati mappati impostando il sistema di coordinate EPSG:32632 - WGS 84 / UTM zone 32N e utilizzando il tool *Geocode Tools* del plugin MMQGIS, che permette di importare file CSV (*Comma-Separated Value*) e attribuire a ciascun indirizzo un punto nello spazio di riferimento. È stata quindi eseguita una correzione della geolocalizzazione di alcuni immobili, specialmente quelli situati nelle zone più rurali, così che ogni punto cadesse con precisione all'interno dei poligoni che delimitano i rispettivi edifici.

Una volta geolocalizzati i dati degli immobili impattati dall'alluvione, è stato possibile calcolare, per ognuno di essi, la minima distanza dall'argine del fiume, il dislivello tra l'edificio e l'argine più vicino, l'area e il numero degli edifici compresi in un raggio di 100mt da quello colpito.

Fatte queste operazioni, si è ottenuto il dataset come riportato in Tabella 2.1.

Importante è sottolineare come in questo studio sia stato preso in esame un set di variabili diverse da quelle considerate in Carisi et al. (2018). Infatti alcune sono state aggiunte (distanza dal fiume, dislivello rispetto l'argine, area e numero degli edifici intorno a quello colpito) mentre altre non sono state considerate (velocità dell'acqua, durata evento).

Variabile	Scheda danno	SWAM	GIS	OMI	Range
Tipologia strutturale immobile	✓			-	
Superficie immobile (m ²)	✓			4 – 1.200	
Valore immobile (€)			✓	4.900 – 1.410.000	
Ammontare del danno (€)	✓			0 – 150.000	
Altezza acqua in casa (mt)		✓		0 – 3,28	
Distanza dal fiume Secchia (mt)			✓	282 – 7.742	
Dislivello rispetto l'argine (mt)			✓	7,56 – 16,70	
Area edifici in un raggio di 100mt da quello colpito (m ²)		✓		0 – 4.827	
Numero edifici in un raggio di 100mt da quello colpito			✓	0 – 17	

Tabella 2.1. Variabili dataset

Capitolo 3

Analisi e Risultati

In questo capitolo saranno affrontate le varie parti dell’analisi eseguita riprendendo quanto visto nel capitolo 2 e adattando quegli argomenti al caso specifico dell’alluvione che ha colpito i comuni di Bastiglia e Bomporto. Tutte le analisi sono state elaborate tramite il software statistico *R* e pacchetti e funzioni a cui si è fatto ricorso saranno nominati quando e se necessario nella forma *pacchetto::funzione*.

Il processo di analisi ha avuto inizio con una prima fase di tipo esplorativo in cui si è fatto ricorso a strumenti di statistica descrittiva.

Considerando la differenza di unità osservate nei due comuni, dalla Figura 3.1 si può intuire la morfologia del territorio e come questa non sembri incidere in modo significativo sul danno relativo degli edifici. Osservando il boxplot del dislivello (c) è chiaro come il Comune di Bomporto sia più basso di quello di Bastiglia in termini di metri s.l.m. (un valore positivo indica una quota minore rispetto l’argine del fiume Secchia) e come, pur trovandosi più lontano dalle sponde del fiume Secchia (d), abbia misurato una maggiore altezza dell’acqua (b). Nonostante ciò la variabile del danno, riscalata per motivi grafici tramite metodo min-max, assume una distribuzione simile per entrambi i comuni (a).

Da una prima analisi descrittiva delle diverse variabili, sia aggregate che disaggregate per comune, non risultano caratteristiche particolari o necessarie di approfondimenti per possibili distorsioni sugli obiettivi di questo studio. Da notare solo la presenza di un’asimmetria a destra per le variabili superficie immobile, valore immobile, ammontare del danno, come era facile aspettarsi.

Risulta però importante, anche per un eventuale impatto sugli algoritmi utilizzati nei passi successivi, il risultato restituito dall’analisi di correlazione e riportato nel corplot (Figura 3.2). I valori riportati corrispondono all’indice di correlazione di Pearson (ρ) e si riferiscono, quindi, a una misura della relazione lineare tra le variabili. Dal grafico si nota come non siano presenti forti relazioni tra le variabili, incluse quelle per cui si può presupporre un legame, come l’altezza dell’acqua e il danno subito ($\rho = -0.01$). I valori più alti sono stati osservati per quelle variabili che hanno un legame logico, quali superficie dell’immobile - valore dell’immobile ($\rho = 0.88$) o dislivello rispetto il fiume Secchia - distanza dal fiume Secchia, per cui si ha $\rho = 0.62$, in linea con quanto analizzato nel boxplot 3.1.

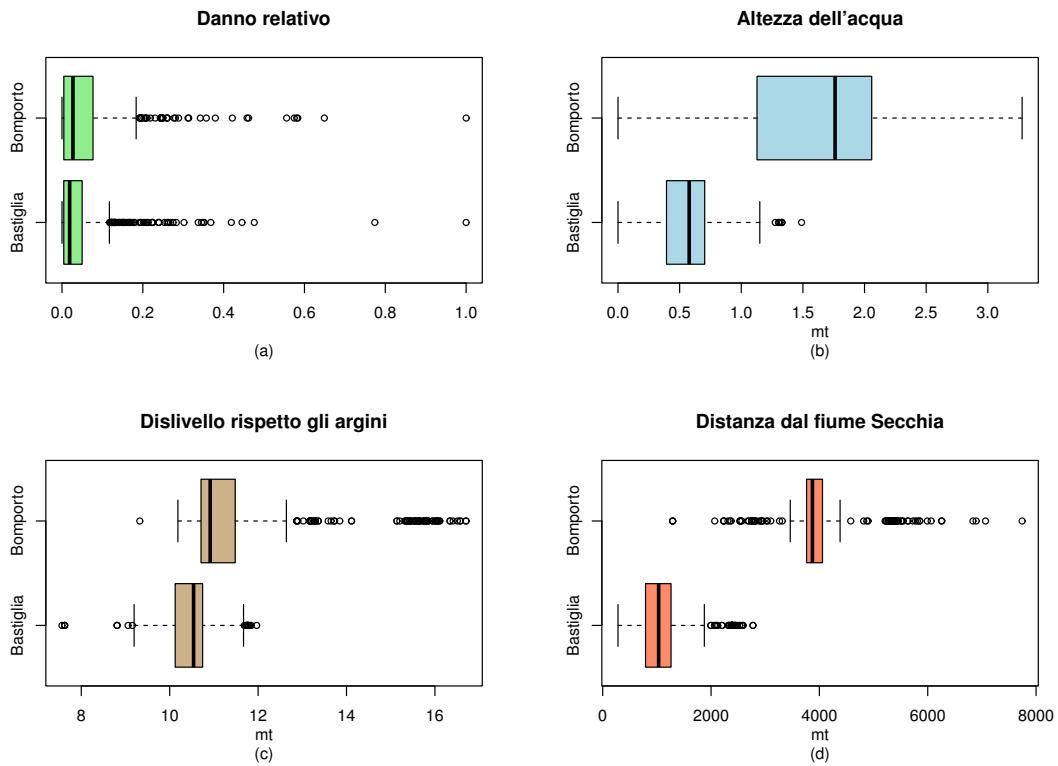


Figura 3.1. Distribuzione danno relativo, altezza dell'acqua, dislivello, distanza dall'argine più vicino

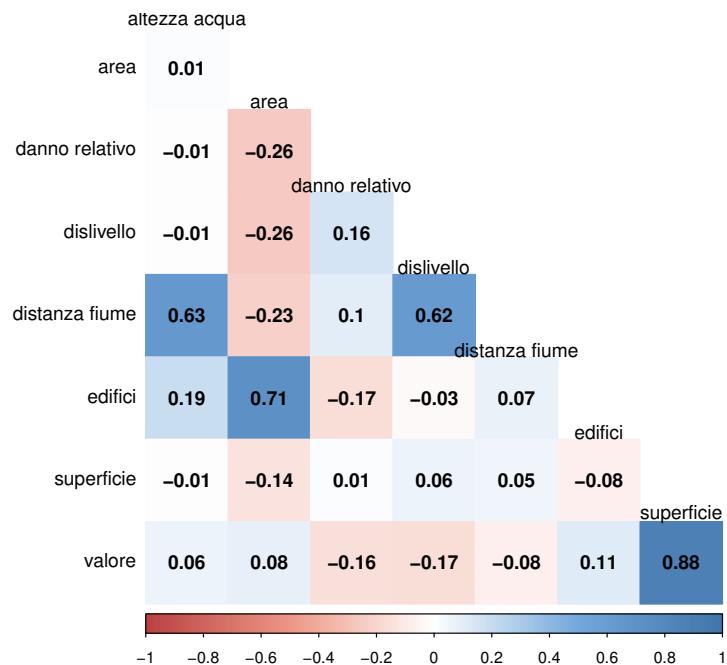


Figura 3.2. Corplot

3.1 La Preparazione dei Dati

Come accennato nella Sezione 2.5.1, uno dei passi più delicati nell’analisi è la preparazione dei dati per le analisi successive, ossia il controllo dell’accuratezza dei dati e l’individuazione di valori mancanti e anomali.

3.1.1 Valori Mancanti

La prima parte del data cleaning riguarda i valori mancanti.

Essendo il dato raccolto a fini di rimborso del danno, difficilmente si possono pensare come causa della mancanza di informazioni le altre domande del questionario o questioni private non considerate. Causa ragionevole di queste assenze, invece, può essere vista nella raccolta e gestione dei dati, e quindi riportata a un evento casuale. Per tali considerazioni questi valori mancanti si fanno ricadere nella categoria MCAR di Little & Rubin (1987).

Sono stati quindi eliminati tutti quei record che riportavano almeno un valore nullo in una delle variabili necessarie all’analisi, riducendo il dataset a 1.366 unità, 886 nel Comune di Bastiglia, 480 a Bomporto. Nonstante questa riduzione nel campione osservato, questo rimane uno dei dataset italiani più completi riguardo i danni da alluvione.

3.1.2 Valori Anomali

Eliminati i valori mancanti, il lavoro si è concentrato sull’individuazione di quei record che si discostavano da valori “normali”.

La presenza di valori anomali si può far ricadere in due casistiche:

1. Provenienti da dichiarazioni non veritieri: i questionari sono stati compilati da normali cittadini e non da personale tecnico atto alla quantificazione di quanto richiesto dalle autorità e questo può portare alla rendicontazione involontaria di misure fuori scala;
2. Presenza di edifici che si discostano realmente dalla “media”.

Per le finalità dell’analisi non si è ritenuto necessario effettuare questo distinguo (comunque di non facile risoluzione) e si è dunque ricorsi a un’analisi multivariata dei valori anomali tramite il calcolo della distanza di Mahalanobis.

Trasformata la variabile della tipologia strutturale in binaria, è stata utilizzata la funzione *stats::mahalanobis* per il calcolo della distanza ed è stato scelto un valore di cutoff pari a $\chi^2_{13,0.99} = 27,68$, dove i gradi di libertà corrispondono al numero di variabili (13) e il livello di significatività α è stato fissato a 0,01 per cercare di selezionare solamente gli outlier più distanti dal resto della distribuzione (un valore $\alpha = 0,05$ avrebbe classificato più unità come valori anomali, e, dato l’obiettivo dello studio e la scarsità di dati, si è deciso di adottare una metodologia più conservativa). La Figura 3.3 mostra il meccanismo di questo metodo in un caso a 3 dimensioni.

Utilizzando questi parametri sono stati trovati 288 dati considerati anomali. Non rientrando negli obiettivi di questo studio, non sono state eseguite analisi più approfondite verso questi valori.

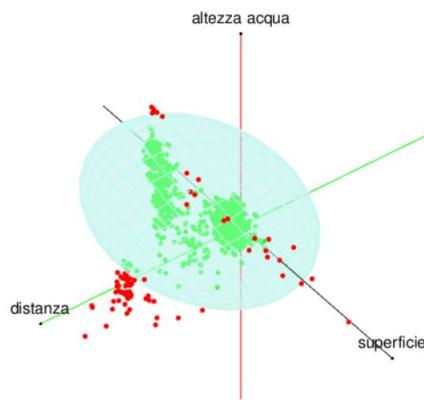


Figura 3.3. Distanza di Mahalanobis

Nonostante siano etichettati come outliers, questi record non sono stati eliminati dal dataset in quanto, testando gli algoritmi supervisionati con e senza questi dati, si è notata una costante diminuzione degli errori di stima nella valutazione di dataset contenenti tali valori. Questi risultati potrebbero essere conseguenza di una scarsa disponibilità di osservazioni in quanto gli algoritmi testati riportano output soddisfacenti con grandi moli di dati. In questo caso, invece, avendo già in origine un dataset ristretto, l'eliminazione di ulteriori 280 record non permetterebbe un buon training del modello.

Al di là di queste osservazioni, i risultati ottenuti sono stati considerati comunque soddisfacenti per gli scopi di questo lavoro e analisi più approfondite riguardo tali valori saranno implementate in studi futuri.

3.2 Apprendimento Supervisionato

Nello studio di Carisi et al. (2018) si riprendono modelli utilizzati in studi di eventi alluvionali avvenuti in varie zone d'Europa senza però ottenere buoni risultati, causa la sitospecificità dell'evento. Si concentra quindi sullo sviluppo di un modello di regressione random forest che, paragonato ai modelli degli altri studi, restituisce buoni risultati.

Avendo osservato la distribuzione delle variabili e i legami esistenti tra di esse, il passo successivo di questo lavoro è stato concentrarsi sull'obiettivo cardine: capire se, e in che ordine di grandezza, le nuove variabili hanno un impatto sugli algoritmi utilizzati nei precedenti lavori su questo specifico evento.

Sono stati presi in considerazione gli algoritmi random forest e xgboost e, trovati i modelli con i parametri migliori, ne sono stati confrontati gli output. Sono stati testati anche modelli di reti neurali ma, avendo queste bisogno di maggior attenzione e tempo, si è deciso di escluderle da questo lavoro. È intenzione del team riprenderle in un approfondimento futuro.

#	% Train	Outliers	ntree	nodesize	mtry	Folds CV	RMSE		R ²	
							Train	Test	Train	Test
1	80%	✓	500	5	3	-	0,09	0,13	0,84	0,50
2	80%		500	5	3	-	0,09	0,14	0,86	0,10
3	70%	✓	500	5	3	-	0,09	0,14	0,84	0,41
4	70%		500	5	3	-	0,09	0,16	0,85	0,25
5	80%	✓	500	5	3	5	0,10	0,13	0,81	0,50
6	80%		500	5	3	5	0,10	0,14	0,82	0,10
7	70%	✓	500	5	3	5	0,11	0,14	0,78	0,42
8	70%		500	5	3	5	0,10	0,16	0,80	0,24
9	80%	✓	200	3	2	5	0,08	0,12	0,88	0,52
10	80%		100	1	2	5	0,09	0,14	0,86	0,10
11	70%	✓	300	6	2	5	0,11	0,14	0,77	0,42
12	70%		100	7	2	5	0,11	0,16	0,75	0,23
13	85%	✓	600	8	2	5	0,10	0,12	0,78	0,64

Tabella 3.1. Alcuni valori di performance modelli testati - Random Forest

3.2.1 Random Forest

Per la risoluzione del modello random forest sono stati utilizzati i pacchetti *randomForest* (<https://cran.r-project.org/web/packages/randomForest>) e *caret* (<https://cran.r-project.org/web/packages/caret/>).

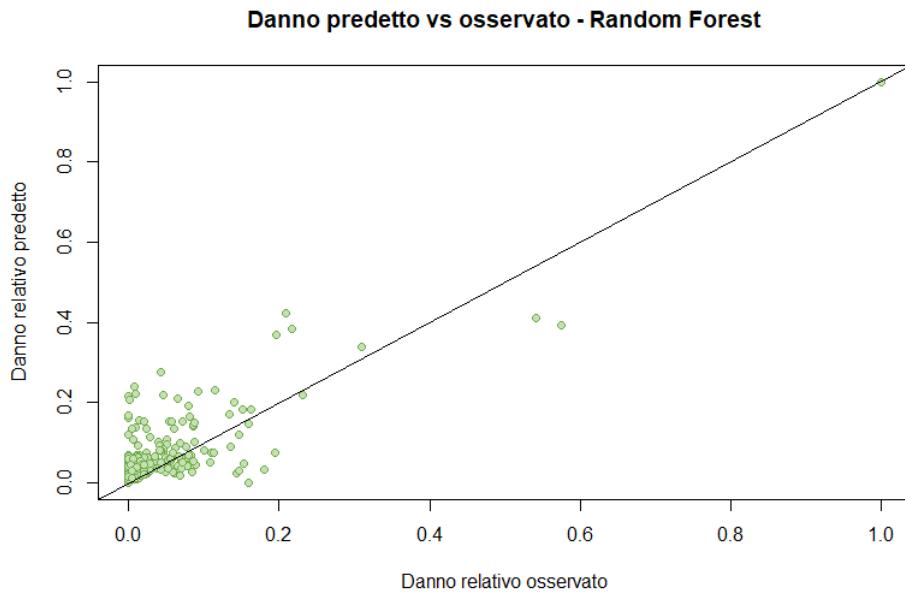
Non avendo problemi nella gestione di variabili categoriche, non è stato necessario trasformare la tipologia strutturale degli immobili in una variabile dummy. Per trovare i migliori parametri di costruzione del modello (sia iperparametri che dimensione train set) sono stati testati diversi modelli, in ognuno dei quali sono stati cambiati uno o più parametri (Tabella 3.1). Più precisamente i vari test si diversificavano per:

- Train set pari a 0,70, 0,80 o 0,85;
- Presenza/Assenza di outliers;
- Numero di alberi da sviluppare;
- Numerosità delle foglie;
- Numero di variabili campionate a ogni split;
- Presenza/Assenza di cross-validation (CV).

In accordo coi risultati di performance in termini di RMSE = $\sqrt{\sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n}}$ e $R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$ (dove y_i valore osservato, \hat{y}_i valore predetto, \bar{y} media valori osservati, n totale delle osservazioni), si è deciso di utilizzare l'85% delle unità come training set e il restante 15% come test set.

Utilizzando la funzione *caret::trainControl* è stato scritto il codice per eseguire un'operazione di k-fold cross-validation con 5 folds e impostando una ricerca randomica su griglia dei parametri. È stata quindi creata la griglia di valori per ntree e nodesize. Infine è stato impostato un ciclo *for* e tramite la funzione *caret::train* è stato eseguito il training del modello. Con un algoritmo così costruito, il tempo di

Set	RMSE	R ²
Train	0,10	0,78
Test	0,12	0,64

Tabella 3.2. Performance random forest**Figura 3.4.** Random forest - valori osservati vs predetti

esecuzione è risultato essere di circa 50 minuti. I valori ottenuti per gli iperparametri sono:

- Numero di alberi (parametro *ntree*): 600;
- Numerosità minima delle foglie (parametro *nodesize*): 8;
- Numero di variabili campionate a ogni split (parametro *mtry*): 2.

Con questo modello si ottengono, come risultati di performance, i valori riportati in Tabella 3.2. Inoltre il grafico dei valori osservati contro quelli predetti (Figura 3.4) restituisce una rappresentazione in linea con quanto presente in letteratura (secondo quanto riassunto in Carisi et al., 2018, tali modelli ottengono valori RMSE in un range 0,062 – 0,13).

3.2.2 XGBoost

Dati i risultati abbastanza soddisfacenti del random forest, si è deciso di utilizzare anche un modello di boosting, e per le caratteristiche dello studio è stato scelto xgboost. In questo caso i pacchetti R utilizzati sono stati *xgboost* (<https://cran.r-project.org/web/packages/xgboost/index.html>) e *caret*.

#	% Train	Outliers	eta	gamma	sub sample	max depth	Folds CV	RMSE		R^2	
								Train	Test	Train	Test
1	80%	✓	0,05	0,03	1	6	-	0,08	0,13	0,82	0,51
2	80%		0,05	0,03	1	6	-	0,08	0,15	0,86	0,05
3	70%	✓	0,05	0,03	1	6	-	0,08	0,14	0,83	0,44
4	70%		0,05	0,03	1	6	-	0,08	0,16	0,85	0,29
5	80%	✓	0,02	0	0,7	4	5	0,10	0,13	0,73	0,52
6	80%		0,02	0	0,7	4	5	0,11	0,15	0,72	0,09
7	70%	✓	0,02	0	0,7	4	5	0,10	0,13	0,76	0,43
8	70%		0,02	0	0,7	4	5	0,10	0,16	0,72	0,28
9	85%	✓	0,04	0	0,7	4	5	0,10	0,11	0,75	0,67

Tabella 3.3. Alcuni valori di performance modelli testati - XGBoost

Set	RMSE	R^2
Train	0,10	0,75
Test	0,11	0,67

Tabella 3.4. Performance xgboost

Come per il random forest, anche per xgboost sono stati testati molteplici modelli con input diversi e il migliore è stato scelto basandosi sui valori RMSE e R^2 (Tabella 3.3).

Per facilitare il confronto con il random forest sono state imposte le stesse proporzioni per i dataset di training e test, 85% train, 15% test. A differenza del modello precedente, in questo caso è stato necessario trasformare la tipologia degli immobili in una variabile binaria. Per come è stato sviluppato, infatti, xgboost richiede come input una matrice di dati al fine di ottimizzare la memoria e velocizzare il training.

L'uso della cross validation e del tuning ha modificato i valori dei seguenti parametri di default (cfr. Sez. 2.2.3):

- Learning rate (*eta*): 0,04;
- Profondità massima di ciascun albero (*max_depth*): 4;
- Proporzione del sottocampione dell'istanza di training (*subsample*): 0,7.

Utilizzando questi parametri sono stati ottenute le performance riportate in Tabella 3.4. Riprendendo quanto fatto con il random forest, sono stati plottati i valori osservati contro quelli predetti e il grafico è riportato in Figura 3.5. Anche in questo caso i risultati ottenuti si avvicinano a quanto si trova in letteratura.

3.2.3 Interpretazione e Confronto

Dati gli output dei modelli allenati, sono stati utilizzati i pacchetti *R inTrees* (<https://cran.r-project.org/web/packages/inTrees>) e *xgboost* (<https://cran.r-project.org/web/packages/xgboost/index.html>) per l'interpretazione dei risultati e rendere così più agevole e chiara la lettura di quanto ottenuto.

I due modelli sono stati interpretati tramite l'estrapolazione delle regole decisionali e confrontati ricorrendo ai valori dell'importanza delle variabili e ai grafici a

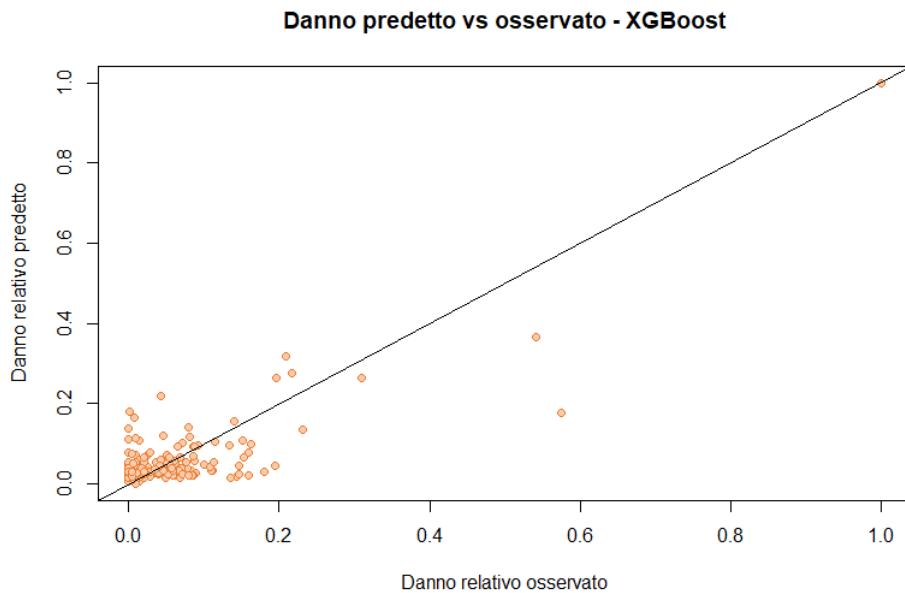


Figura 3.5. *XGBoost - valori osservati vs predetti*

dipendenza parziale. Come misura di confronto sono stati utilizzati anche i valori delle performance in termini di *RMSE* e R^2 riportati nelle sezioni 3.2.1 e 3.2.2.

Regole Decisionali

Con il termine “regola decisionale” si intende un’istruzione del tipo *if - then* per cui *if* detta quelle condizioni che, se soddisfatte, restituiscono in output quanto afferma *then*. La struttura di un albero decisionale (e suoi derivati) permette l’applicazione di una logica simile che può essere quindi utilizzata per capire come sono costituiti i modelli sviluppati.

A questo scopo è stato utilizzata la funzione *inTrees:getRuleMetric*. A titolo esemplificativo si riporta in Tabella 3.5 un estratto di quanto ottenuto, e nello specifico le voci riportate corrispondono a:

- Condizione: combinazione delle regole decisionali (più di 10.000 per random forest e più di 2.000 per xgboost) utilizzate dagli alberi che costituiscono il modello finale;
- Previsione: media degli output di quei record che soddisfano la Condizione;
- Frequenza: percentuale di unità che soddisfano la condizione, chiamata anche “popolarità della condizione”.

Disponendo di questo elenco di regole i modelli elaborati diventano più comprensibili in quanto la struttura delle condizioni è di più facile interpretazione. Si può infatti affermare che il 78% dei record collezionati risponde alla prima condizione e un nuovo record che soddisfa tale regola restituisca una previsione di danno relativo pari a 0,08, ossia un danno assoluto pari all’8% del valore dell’immobile.

Modello	Condizione	Previsione	Frequenza
Random Forest	distanza $\leq 4360\text{mt}$ & dislivello $\leq 11\text{mt}$ & area degli edifici $> 495\text{m}^2$ & n. edifici > 2 & valore dell'immobile $> 53.400\text{\euro}$	0,08	78%
	$467\text{m}^2 < \text{area degli edifici} \leq 610\text{m}^2$ & $39.375\text{\euro} < \text{valore dell'immobile} \leq 115.875\text{\euro}$	0,27	1%
XGBoost	superficie immobile $> 20\text{ m}^2$ & distanza dal fiume $\leq 4290\text{mt}$ & dislivello $\leq 13\text{mt}$ & area degli edifici $> 311\text{m}^2$	0,08	87%
	$0,2\text{mt} < \text{altezza dell'acqua} \leq 0,3\text{mt}$ & area degli edifici $> 330\text{m}^2$ & valore dell'immobile $\leq 99.662\text{\euro}$	0,27	1%

Tabella 3.5. Esempio regole decisionali ottenute dai modelli

Feature Importance

Per estrapolare i valori di feature importance sono state utilizzate le funzioni *caret::varImp* per il random forest e *xgboost::xgb.importance* per xgboost. Restituendo le due funzioni valori su misurazioni e ordini di grandezza differenti è stato necessario, ai fini di un confronto, eliminare la tipologia dell'edificio e riscalare i dati tramite, in questo caso, la trasformazione *min-max*. Il risultato è in Figura 3.6.

Guardando anche i grafici precedenti, si nota come per entrambi i modelli la tipologia strutturale degli immobili e l'altezza dell'acqua siano variabili poco significative dal punto di vista previsionale. La prima può trovare giustificazione nell'assenza, in questa analisi, di edifici con strutture considerate deboli, come le costruzioni in legno. L'altezza dell'acqua, invece, è sempre stata parametro centrale degli studi inerenti la stima dei danni causati da fenomeni alluvionali, dando poca importanza alle variabili incluse in questo lavoro. Quanto ottenuto porta a considerare l'importanza di ampliare lo spettro delle variabili da prendere in esame per questa tipologia di eventi.

Dipendenza Parziale

Un ultimo confronto tra i due modelli è stato effettuato utilizzando i grafici a dipendenza parziale, rivelatisi interessanti per indirizzare futuri approfondimenti. Per ottenere i grafici pdp sono state utilizzate le funzioni *iml::FeatureEffect* per random forest e *pdp::partial* per xgboost. Per una più corretta lettura dei risultati, i *pdp* sono accompagnati dalla distribuzione di densità della variabile presa in considerazione.

In Figura 3.7 sono riportati, come esempio, i grafici relativi alla variazione dell'output in risposta a una variazione delle variabili “altezza dell'acqua”, “dislivello”, “numero edifici intorno a quello colpito” e “area degli edifici attorno a quello colpito”.

Dalla Figura 3.7a si evidenzia come, in media, il danno relativo tenda a crescere fino a un livello dell'acqua pari a circa 1 metro, andando quindi a stabilizzarsi per poi riprendere a crescere nel range 2,0 – 2,5 mt. Si può inoltre osservare come nel range 0,0 – 0,5 mt sia presente una decrescita del danno, contro la logica dell'evento. Tale risultato può essere dato da diversi elementi di disturbo, come la correttezza del dato di input, ma, esulando dagli obiettivi di questo lavoro, la causa di questo

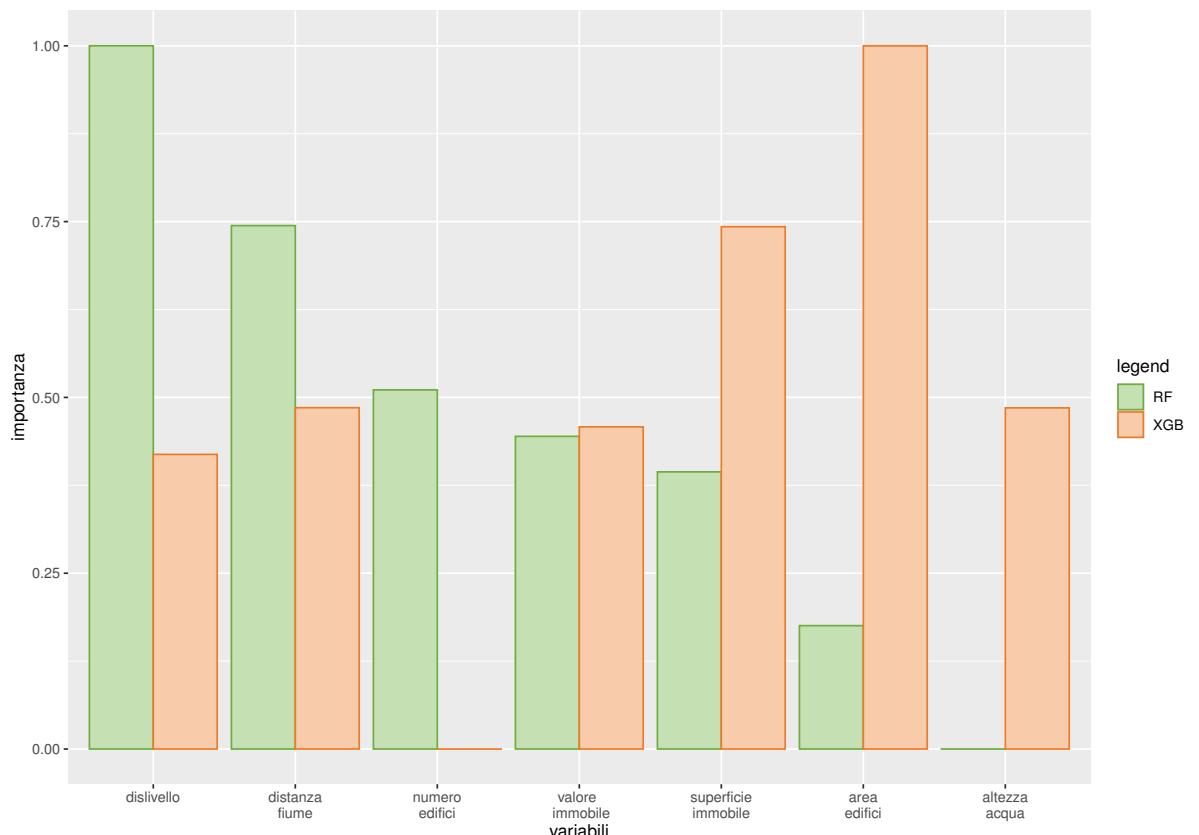
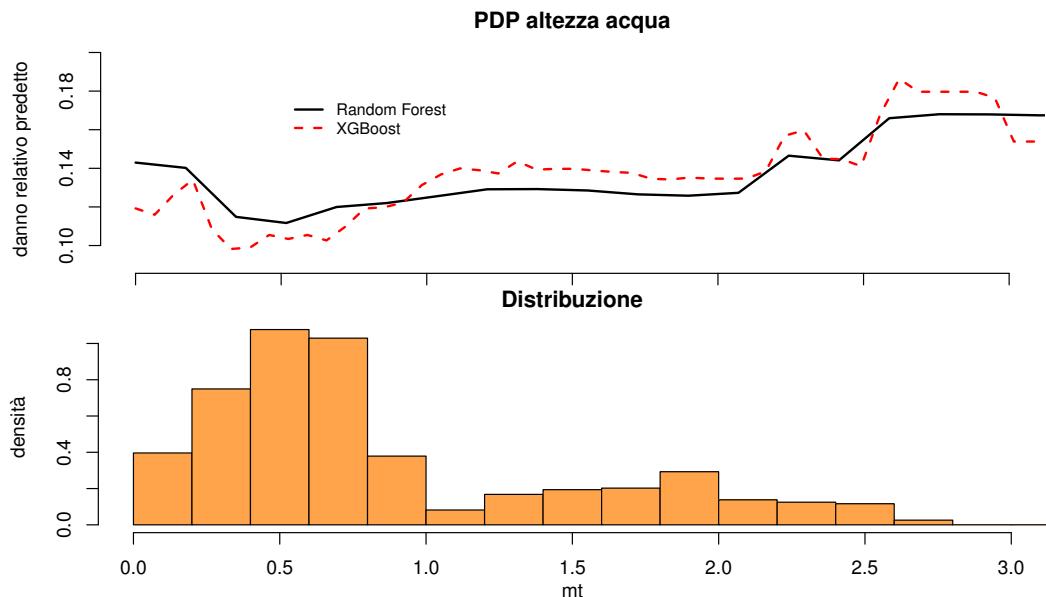
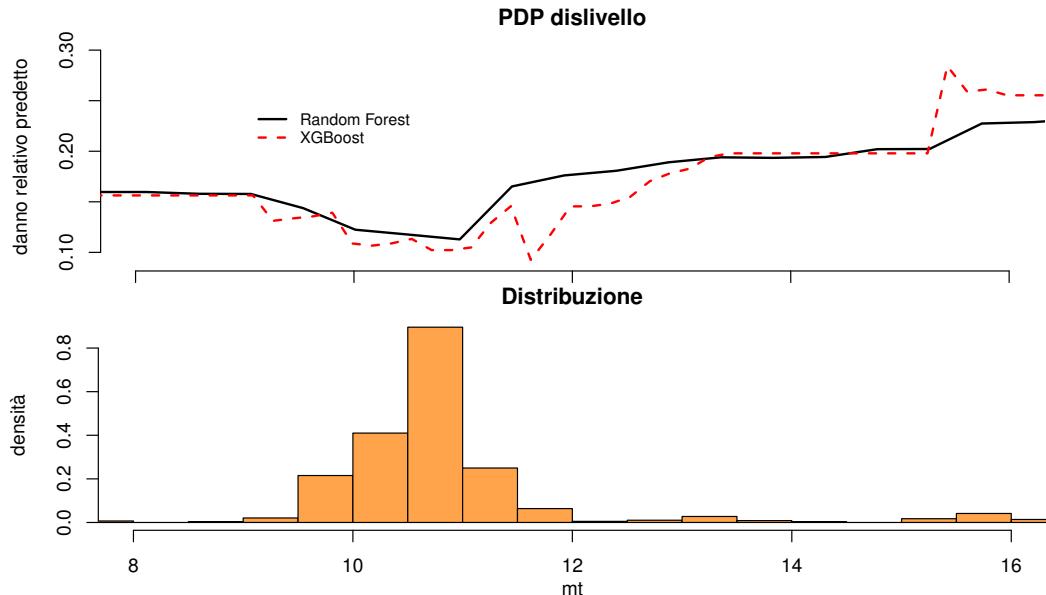


Figura 3.6. Confronto feature importance

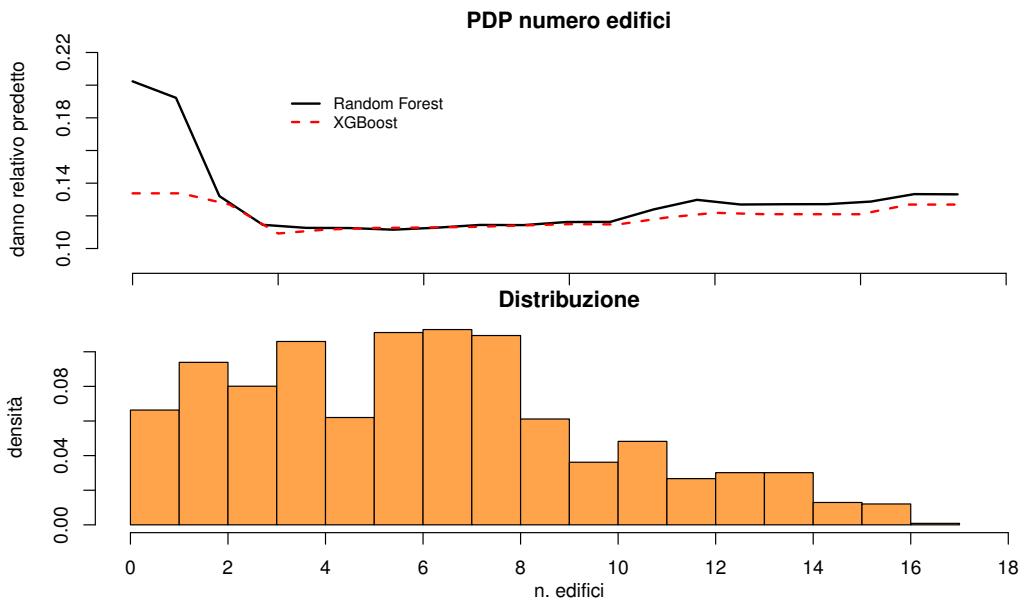


(a) Danno relativo predetto vs altezza dell'acqua e Distribuzione altezza dell'acqua

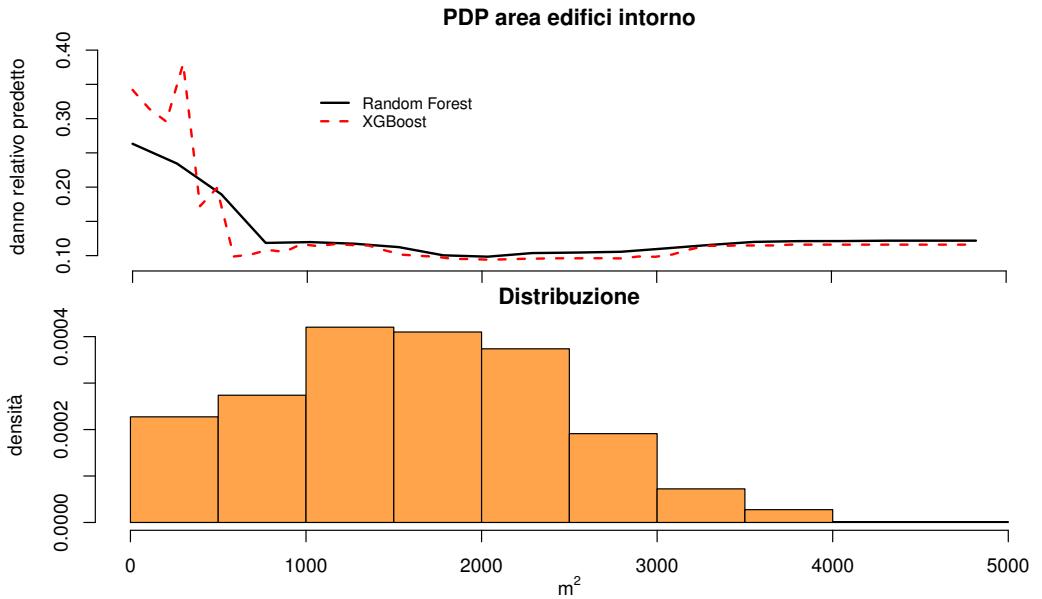


(b) Danno relativo predetto vs dislivello e Distribuzione dislivello

Figura 3.7. Grafici di dipendenza parziale



(c) Danno relativo predetto vs numero edifici e Distribuzione numero edifici



(d) Danno relativo predetto vs area edifici intorno e Distribuzione area edifici

Figura 3.7. Grafici di dipendenza parziale

disallineamento rispetto le aspettative sarà oggetto di approfondimenti futuri. In generale si può comunque osservare un trend crescente, pur non essendoci una relazione lineare.

Dal pdp del dislivello (Figura 3.7b) si osserva come il danno relativo tenda a decrescere fino a un dislivello pari a circa 11 metri per poi tornare ad avere una tendenza crescente. Ricordiamo che un dislivello positivo indica una posizione inferiore dell'immobile rispetto agli argini del fiume Secchia.

Interessante risulta la Figura 3.7c. Si nota come il numero di edifici impatti positivamente nella riduzione del danno in quanto questo diminuisce all'aumentare del numero di edifici presenti in un raggio di 100 mt attorno a quello colpito. Nonostante si potesse assumere questo ragionamento vero sul piano empirico, è stato interessante per il team trovare riscontro nei dati osservati.

Il grafico risulta concorde all'andamento relativo degli edifici intorno a quello colpito (Figura 3.7d), il che rafforza l'idea che l'avere altre strutture intorno ha aiutato nella riduzione del danno registrato in questo specifico evento.

Nonostante in ciascun grafico siano presenti delle tendenze, è da porre attenzione nei range dei valori per cui si hanno pochi dati osservati in quanto, essendo i risultati della dipendenza parziale delle medie (come spiegato in 2.2.4), in queste zone la lettura degli output può risultare distorta e quindi meno affidabile.

3.3 Apprendimento non Supervisionato

Analizzato il comportamento del danno in funzione delle altre variabili osservate, si è deciso di proseguire le analisi utilizzando metodologie della cluster analysis per cercare e analizzare eventuali gruppi omogenei di unità.

Non essendo presenti in letteratura riferimenti utili all'obiettivo di questa Sezione e non potendo impostare a priori un determinato numero di cluster, in questa fase è stato utilizzato un algoritmo gerarchico di tipo agglomerativo.

Prima di procedere con l'algoritmo vero e proprio, tramite la funzione *scale* di R sono stati standardizzati i dati ed è stata calcolata la matrice delle dissimilarità utilizzando la funzione *stats:dist* impostando la distanza euclidea come misura. È stato quindi possibile calcolare le distanze tra i cluster ricorrendo a *stats::hclust* e utilizzando la distanza di Ward. Rappresentato lo screeplot (Figura 3.8) e utilizzando il metodo del gomito, è stato deciso di utilizzare un numero di cluster pari a 8, ottenendo il dendrogramma in Figura 3.9. Osservando lo screeplot è possibile notare come sia corretta anche una scelta di 5, 10 o 12 cluster ma per mantenere un corretto rapporto tra raggruppamento e lettura dei risultati sono stati scelti gli 8 cluster visibili nel dendrogramma.

Utilizzando la funzione *factoextra::elbow* è stata quindi creata la suddivisione e ne è stata verificata la bontà tramite il calcolo e la rappresentazione della silhouette, riportata in Figura 3.10. Da questa si nota la presenza di unità con valori negativi, e quindi non ben clusterizzati. Come soluzione è stato deciso di spostare queste osservazioni nel cluster più vicino. In Tabella 3.6 sono riportate le medie delle variabili di ogni cluster e la percentuale di outlier in ognuno di essi.

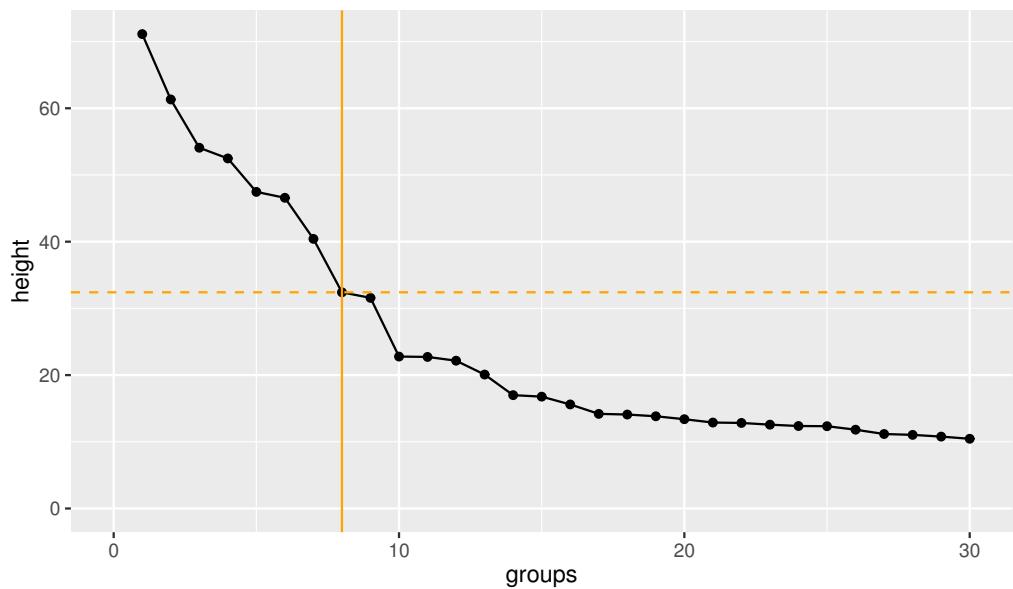


Figura 3.8. Screeplot

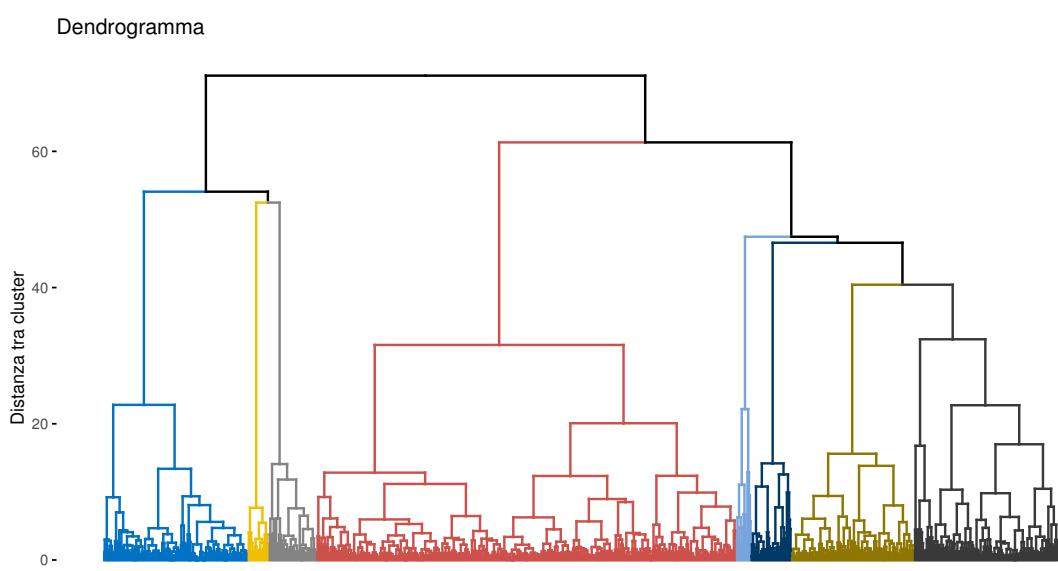
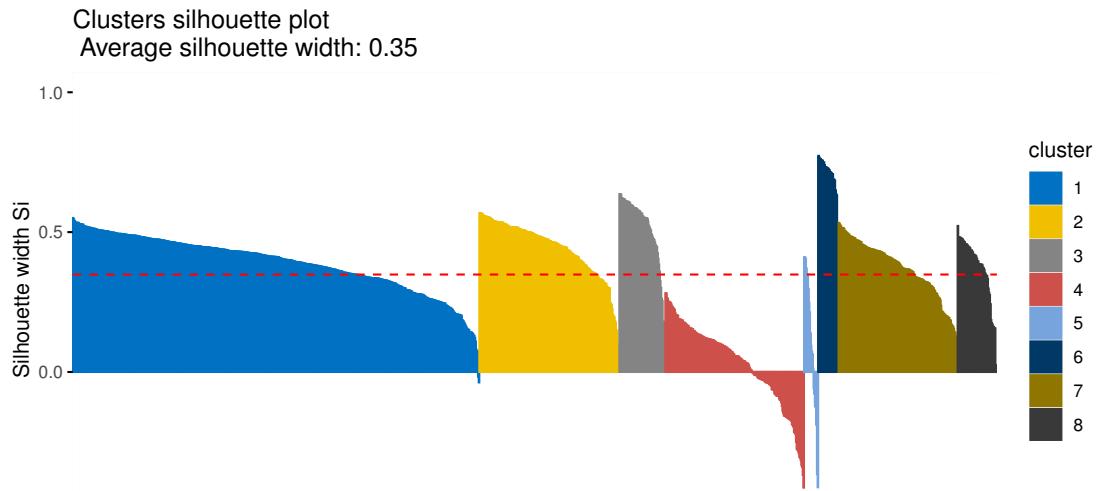


Figura 3.9. Dendrogramma

**Figura 3.10.** Silhouette

Cluster	1	2	3	4	5	6	7	8
Numerosità	617	208	68	129	15	30	238	61
% Tipo altro	0%	0%	100%	0%	0%	0%	0%	3%
% Tipo cemento	0%	100%	0%	0%	20%	0%	0%	10%
% Tipo cemento e muratura	0%	0%	0%	0%	7%	100%	0%	0%
% Tipo muratura	100%	0%	0%	100%	73%	0%	100%	87%
Superficie immobile (m ²)	105	90,70	118	135	560	101	103	140
Altezza dell'acqua (mt)	0,52	0,86	0,75	1,40	1,18	0,50	1,87	0,27
Distanza dal fiume (mt)	1.028	1.894	1.505	3.370	2.086	1.060	3.879	5.485
Dislivello (mt)	10,40	10,60	10,60	11,30	10,60	10,40	10,90	15,70
Area edifici intorno (m ²)	1.728	1.638	1.840	361	1.748	1.610	1.867	741
N. edifici intorno	7	6	7	2	8	6	9	6
Valore immobile (€)	116.638	105.583	132.880	84.319	667.620	108.457	125.399	63.015
Danno relativo (€)	0,09	0,06	0,09	0,34	0,02	0,07	0,08	0,19
% Outlier	7%	50%	53%	26%	80%	30%	9%	48%

Tabella 3.6. Analisi Cluster



Figura 3.11. Geolocalizzazione dei cluster

Variabile	Varianza nei cluster (WSS)	Varianza tra cluster (GSS)	Pseudo-F
Distanza dal fiume	286	1.079	731
Dislivello	293	1.072	711
Altezza dell'acqua	548	817	289
Valore immobile	726	639	171
Superficie	822	543	128
Area edifici intorno	964	401	81
N. edifici intorno	1.034	331	62
Danno relativo	1.131	234	40

Tabella 3.7. Valori pseudo-F

Per avere una visuale georeferenziata di quanto ottenuto, tramite il software QGIS sono stati plottati i cluster con una scala di colori basata sul danno medio. Il risultato è raffigurato in Figura 3.11.

Come ultima analisi è stato calcolato l'indice *Pseudo-F* per visualizzare il contributo dato da ogni singola variabile alla classificazione. I valori sono riportati in Tabella 3.7 in ordine decrescente. L'indice è calcolato come $\frac{GSS/K-1}{WSS/N-K}$, dove GSS è la varianza tra i cluster mentre WSS è quella interna agli stessi, K rappresenta il numero di cluster e N il totale delle osservazioni. Essendo l'obiettivo della cluster analysis quello di massimizzare la varianza inter-cluster e minimizzare quella intra-cluster, valori alti per la pseudo-F indicano cluster densi e distanti. Se il valore decresce significa che la varianza all'interno dei cluster sta aumentando o la varianza tra i cluster sta diminuendo.

Dai valori riportati nella tabella si nota come le variabili “distanza dal fiume” e “dislivello” siano quelle che hanno caratterizzato maggiormente la clusterizzazione, restituendo valori di pseudo-F maggiori rispetto alle altre variabili. Da quanto trovato, si può inoltre dire che, per i dati a disposizione, gli edifici sono stati suddivisi

principalmente in base alla loro posizione (data dalle variabili “distanza dal fiume” e “dislivello”) e che le caratteristiche urbanistiche (“valore dell’immobile”, “superficie immobile”, “area edifici intorno”, “numero edifici intorno”) non hanno avuto, nel processo, un’influenza così importante come si poteva essere portati a pensare a priori. Da questa osservazione si può ipotizzare di forzare la clusterizzazione escludendo le variabili topografiche per osservare quali siano le variabili strutturali maggiormente impattanti nella ridefinizione dei nuovi cluster.

Capitolo 4

Conclusioni e Sviluppi Futuri

Nato con l'obiettivo di studiare l'alluvione avvenuta nel 2014 nei comuni di Bastiglia e Bomporto, il lavoro presentato mostra aspetti interessanti del fenomeno e si pone come punto di partenza per futuri approfondimenti.

Importanti sono i risultati ottenuti dai modelli ad albero. I risultati ottenuti in Sez. 3.2.1 e 3.2.2 è da considerarsi soddisfacente per gli scopi di questo lavoro in quanto, tramite la costruzione di modelli multivariati di facile interpretazione, si è riusciti ad avere una buona rappresentazione del danno osservato e un errore di stima accettabile. È da tenere in considerazione, infatti, che in letteratura non sono presenti benchmark per uno studio così fatto, poichè lavori su eventi simili sono stati effettuati per altre località europee (Spekkers et al., 2014; Nafari et al., 2017) ma questa tipologia di fenomeni, e quindi di modelli, soffrono di sitospecificità. Inoltre studi simili a questo, pur utilizzando modelli ad albero, sono basati su set di variabili diverse. Interessante è stata anche l'introduzione dei grafici di dipendenza parziale con cui si è riusciti a ottenere una visione più chiara di come il danno può essere influenzato dalle altre variabili.

L'utilizzo del clustering in Sez. 3.3 pone le basi per futuri approfondimenti con un nuovo approccio. Utili risultano le rappresentazioni dei cluster su mappa e il calcolo del valore della pseudo-F, elementi che apportano maggior comprensione sulla distribuzione delle caratteristiche del fenomeno e nuove considerazioni in merito alle variabili.

Le analisi effettuate e esposte in questo lavoro dimostrano come sia importante l'approccio di nuove tecniche in un campo come quello dello studio degli eventi alluvionali, sempre più presenti e importanti all'interno del più vasto insieme dei fenomeni naturali.

Le osservazioni esposte in questo lavoro, quindi, non sono da considerarsi come definitive ma anzi sono utili per individuare le strategie da adottare e le fasi da approfondire in studi futuri.

Ruolo centrale assume anche la raccolta e la gestione dei dati. Il sistema con cui oggi si reperiscono e si conservano introduce fattori di errore, parzialmente evitabili con l'introduzione di nuovi processi e infrastrutture gestionali. Questo è un punto particolarmente importante in quanto senza una buona struttura del dato alla base non è possibile sfruttare appieno la potenza dei modelli adottati, e l'impegno necessario a soddisfare questo obiettivo sarebbe ampiamente ripagato

con lo sviluppo di metodi e algoritmi più efficienti, e quindi un miglior approccio a questa tipologia di eventi.

Essendo questo un punto di partenza per ulteriori approfondimenti, i principali futuri sviluppi riguarderanno:

- uno studio specifico sui valori anomali e il loro trattamento
- miglioramento dei grafici di dipendenza parziale con anche la realizzazione di *Accumulated Local Effect plot* (basati sulla distribuzione condizionata)
- noti i risultati della pseudo-F, estrapolazione di maggiori informazioni dalla clusterizzazione
- Utilizzo di metodologie più complesse, quali reti neurali o algoritmi di apprendimento semi-supervisionato.

Elenco delle figure

2.1	<i>Boxplot ([74])</i>	10
2.2	<i>Valori anomali multivariati</i>	11
2.3	<i>Albero Decisionale ([79])</i>	14
2.4	<i>Evoluzione algoritmi ad albero</i>	15
2.5	<i>PDP ([80])</i>	21
2.6	<i>Clustering agglomerativo vs divisivo ([81])</i>	23
2.7	<i>Dendrogramma ([82])</i>	26
2.8	<i>Metodo del gomito ([83])</i>	27
2.9	<i>Zona inondata ([13])</i>	30
2.10	<i>Esempio scheda di danno</i>	31
3.1	<i>Distribuzione danno relativo, altezza dell'acqua, dislivello, distanza dall'argine più vicino</i>	35
3.2	<i>Corplot</i>	35
3.3	<i>Distanza di Mahalanobis</i>	37
3.4	<i>Random forest - valori osservati vs predetti</i>	39
3.5	<i>XGBoost - valori osservati vs predetti</i>	41
3.6	<i>Confronto feature importance</i>	43
3.7	<i>Grafici di dipendenza parziale</i>	44
3.7	<i>Grafici di dipendenza parziale</i>	45
3.8	<i>Screeplot</i>	47
3.9	<i>Dendrogramma</i>	47
3.10	<i>Silhouette</i>	48
3.11	<i>Geolocalizzazione dei cluster</i>	49

Elenco delle tabelle

2.1	Variabili dataset	33
3.1	Alcuni valori di performance modelli testati - Random Forest	38
3.2	Performance random forest	39
3.3	Alcuni valori di performance modelli testati - XGBoost	40
3.4	Performance xgboost	40
3.5	Esempio regole decisionali ottenute dai modelli	42
3.6	Analisi Cluster	48
3.7	Valori pseudo-F	49

Materiale Consultato

Bibliografia

- [1] Amadio *et al.*, “Improving flood damage assessment models in Italy”, in *Natural Hazards*, LXXXII (2016), pp. 2075-2088, doi:<https://doi.org/10.1007/s11069-016-2286-0>
- [2] Apley, D.W., Zhu, J., “Visualizing the effects of predictor variables in black box supervised learning models”, in *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, LXXXII (2020), pp. 1059-1086, doi:<https://doi.org/10.1111/rssb.12377>
- [3] Arrighi *et al.*, “Urban micro-scale flood risk estimation with parsimonious hydraulic modelling and census data”, in *Natural Hazards and Earth System Sciences*, XIII (2013), pp. 1375-1391, doi:<https://doi.org/10.5194/nhess-13-1375-2013>
- [4] Balaji *et al.*, “DeepRacer: Educational Autonomous Racing Platform for Experimentation with Sim2Real Reinforcement Learning”, 2019, doi:<https://doi.org/10.48550/arXiv.1911.01562>
- [5] Barnett, V., Lewis, T., *Outliers in statistical data*, Wiley, 1994³
- [6] Behera, S., Rani, R., “Comparative analysis of density based outlier detection techniques on breast cancer data using hadoop and map reduce”, *2016 International Conference on Inventive Computation Technologies (ICICT)*, 2016, pp. 1-4
- [7] Ben-Gal, I., “Outlier detection”, in *Data Mining and Knowledge Discovery Handbook* (2005), pp. 131-146
- [8] Boehmke, B., Greenwell, B., *Hands-on machine learning with R*, Chapman and Hall/CRC, 2019
- [9] Breiman, *et al.*, *Classification and regression trees*, Chapman and Hall/CRC, 1984
- [10] Breiman, L., “Random Forests”, in *Machine Learning*, XLV (2001), pp. 5-32
- [11] Breunig *et al.*, “LOF: Identifying density-based local outliers”, 2000
- [12] Cameron, A.C., Trivedi, P., *Microeconometrics: methods and applications*, 2005

- [13] Carisi *et al.*, “Development and assessment of uni- and multivariable flood loss models for Emilia-Romagna (Italy)”, in *Natural Hazards and Earth System Sciences*, XVIII (2018), pp. 2057-2079, doi:<https://doi.org/10.5194/nhess-18-2057-2018>
- [14] Chen, T., Guestrin C., “XGBoost: A Scalable Tree Boosting System”, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785-794
- [15] Chinh *et al.*, “Multi-variate analyses of flood loss in Can Tho city, Mekong delta”, in *Water*, VIII (2016)
- [16] Darmaraki *et al.*, “Past variability of Mediterranean Sea marine heat-waves”, in *Geophysical Research Letters*, XLVI (2019), pp. 9813-9823, doi:<https://doi.org/10.1029/2019GL082933>
- [17] Díaz-Uriarte, R., Alvarez de Andrés, S., “Gene selection and classification of microarray data using random forest”, in *BMC Bioinformatics*, 2006
- [18] Domeneghetti *et al.*, “Evolution of flood risk over large areas: Quantitative assessment for the Po river”, in *Journal of Hydrology*, DXXVII (2015), pp. 809-823, doi:<https://doi.org/10.1016/j.jhydrol.2015.05.043>
- [19] Ester *et al.*, “A density-based algorithm for discovering clusters”, 1996
- [20] Filzmoser, P., “A multivariate outlier detection method”, 2004
- [21] Filzmoser *et al.*, “Multivariate outlier detection in exploration geochemistry”, in *Computers & Geosciences*, XXXI (2005), pp. 579-587
- [22] Freund, Y., Schapire, R.E., “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting”, in *Journal of Computer and System Sciences*, LV (1997), pp. 119-139
- [23] Friedman, J.H., “Greedy Function Approximation: A Gradient Boosting Machine”, in *The Annals of Statistics*, XXIX (2001), pp. 1189-1232
- [24] Gauci *et al.*, “Horizon: Facebook’s Open Source Applied Reinforcement Learning Platform”, 2018, doi:<https://doi.org/10.48550/arXiv.1811.00260>
- [25] Géron, A., *Hands-on machine learning with scikit-learn, keras & tensorflow*, O’Reilly Media, 2019²
- [26] Gnanadesikan, R., *Methods for Statistical Data Analysis of Multivariate Observations*, Wiley-Interscience, 1997²
- [27] Goldstein *et al.*, “Random Forests for Genetic Association Studies”, in *Statistical Applications in Genetics and Molecular Biology*, X (2011)
- [28] Goldstein *et al.*, “Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation”, in *Journal of Computational and Graphical Statistics*, XXIV (2015), pp. 44-65, doi:<https://doi.org/10.1080/10618600.2014.907095>

- [29] Graham, J.W., Hofer, S.M., "Multiple imputation in multivariate research", in *Modeling longitudinal and multilevel data: Practical issues, applied approaches, and specific examples*, 2000, pp. 201-218
- [30] Graham *et al.*, "Method for handling missing data", in *Handbook of Psychology: Research methods in psychology*, II (2003), pp. 87-114, doi:<https://doi.org/10.1002/0471264385.wei0204>
- [31] Grissom II *et al.*, "Don't Until the Final Verb Wait: Reinforcement Learning for Simultaneous Machine Translation", in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1342-1352, doi:<http://dx.doi.org/10.3115/v1/D14-1140>
- [32] Grubbs , F.E., "Procedures for detecting outlying observations in samples", 1969
- [33] Hasanzadeh *et al.*, "Flood loss modelling with FLF-IT: a new flood loss function for Italian residential structures", in *Natural Hazards and Earth System Sciences*, XVII (2017), pp. 1047-1059, doi:<https://doi.org/10.5194/nhess-17-1047-2017>
- [34] Hastie *et al.*, *The elements of statistical learning*, Springer, 2017²
- [35] Hawkins, D.M., *Identification of outliers*, Springer, 1980
- [36] Intergovernmental Panel on Climate Change, *Climate Change 2014: Synthesis Report*, 2014, doi: <https://doi.org/10.1017/CBO9781107415324>
- [37] James *et al.*, *An introduction to statistical learning*, Springer, 2021²
- [38] Jin *et al.*, "Real-Time Bidding with Multi-Agent Reinforcement Learning in Display Advertising", in *CIKM '18: Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 2193-2201, doi:<https://doi.org/10.1145/3269206.3272021>
- [39] Johnson, R.A., Wichern, D.W., *Applied Multivariate Statistical Analysis*, Pearson, 2007
- [40] Kassambara, A., *Practical Guide To Cluster Analysis in R - Unsupervised Machine Learning*, STHDA, 2017
- [41] Kaufman, L., Rousseeuw, P.J., "Partitioning Around Medoids (Program PAM)", in *Wiley Series in Probability and Statistics*, 1990, pp. 68-125, doi:<https://doi.org/10.1002/9780470316801.ch2>
- [42] Kiran *et al.*, "Deep Reinforcement Learning for Autonomous Driving: A Survey", in *IEEE Transactions on Intelligent Transportation Systems*, 2021, pp. 1-18, doi:<https://doi.org/10.1109/TITS.2021.3054625>
- [43] Kreibich *et al.*, "Probabilistic, Multivariable Flood Loss Modeling on the Mesoscale with BT-FLEMO", in *Risk Analysis*, XXXVII (2017), pp. 774-787, doi:<https://doi.org/10.1111/risa.12650>

- [44] Li *et al.*, “Deep Reinforcement Learning for Dialogue Generation”, in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1192-1202, doi:<http://dx.doi.org/10.18653/v1/D16-1127>
- [45] Lindholm *et al.*, *Machine Learning - A First Course for Engineers and Scientists*, Cambridge University Press, 2021
- [46] Liu *et al.*, “Isolation forest”, in *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 413-422
- [47] MacQueen, J.B., “Some Methods for classification and Analysis of Multivariate Observations”, in *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, I (1967), pp. 281-297
- [48] Merz *et al.*, “Multi-variate flood damage assessment: a tree-based data-mining approach”, in *Nat. Hazards Earth Syst. Sci.*, XIII (2013), pp. 53-64
- [49] Molinari *et al.*, “Le curve di danno quale strumento a supporto della direttiva alluvioni: criticità dei dati italiani”, in *XXXIII Convegno Nazionale di Idraulica e Costruzioni Idrauliche*, 2012
- [50] Molnar, C., *Interpretable machine learning*, 2020
- [51] Orlandini *et al.*, “Evidence of an emerging levee failure mechanism causing disastrous floods in Italy”, in *Water Resources Research*, LI (2015), pp. 7995–8011, doi:<https://doi.org/10.1002/2015WR017426>
- [52] Pastormerlo, M., *SWAM (Surface Water Analysis Method): Un Metodo Speditivo per la Modellazione di un Evento Alluvionale.*, Master Thesis, Universitá degli Studi di Milano, Milan, Italy, 2016
- [53] Paulus *et al.*, “A Deep Reinforced Model for Abstractive Summarization”, 2017, doi:<https://doi.org/10.48550/arXiv.1705.04304>
- [54] Probst *et al.*, “Hyperparameters and tuning strategies for random forest”, in *WIREs: Data Mining and Knowledge Discovery*, IX (2019)
- [55] Rubin, D., Little, R.J.A., *Statistical analysis with missing data*, 1987
- [56] Schröter *et al.*, “Tracing the value of data for flood loss modelling”, in *FLOODrisk 2016*, VII (2016), doi:<https://doi.org/10.1051/e3sconf/20160705005>
- [57] Scornet *et al.*, “Tuning parameters in random forests”, in *Esaim:Proceedings and Surveys*, LX (2017), pp. 144-162, doi:<https://doi.org/10.1051/proc/201760144>
- [58] Segal, M.R., “Machine Learning Benchmarks and Random Forest Regression”, in *UCSF: Center for Bioinformatics and Molecular Biostatistics*, 2004, <https://escholarship.org/uc/item/35x3v9t4>
- [59] Silver *et al.*, “Mastering the game of Go without human knowledge”, in *Nature*, DL (2017), pp. 354-359, doi:<https://doi.org/10.1038/nature24270>

- [60] Smith, D.I., “Flood damage estimation - a review of urban stage-damage curves and loss functions”, in *Water SA*, XX (1994), pp. 231-238
- [61] Soley-Bori, M., “Dealing with missing data: Key assumptions and methods for applied analysis”, 2013
- [62] Spekkers *et al.*, “Decision-tree analysis of factors influencing rainfall-related building structure and content damage”, in *Natural Hazards and Earth System Sciences*, XIV (2014), pp. 2531-2547, doi:<https://doi.org/10.5194/nhess-14-2531-2014>
- [63] Sutton, R.S., Barto, A.G., *Reinforcement Learning: An Introduction*, MIT Press, 1998
- [64] Thang, T.M, Kim, J., “The Anomaly Detection by Using DBSCAN Clustering with Multiple Parameters”, *2011 International Conference on Information Science and Applications*, 2011, pp. 1-5
- [65] Warren *et al.*, “Use of Mahalanobis distance for detecting outliers and outlier clusters in markedly non-normal data: a vehicular traffic example”, 2011
- [66] Wayman, J., *Multiple imputation for missing data: what is it and how can i use it?*, 2003
- [67] White, G.F., *uman Adjustment to Floods*, 1945
- [68] Yu *et al.*, “Reinforcement Learning in Healthcare: A Survey”, 2019, doi:<https://doi.org/10.48550/arXiv.1908.08796>
- [69] Zhao, Q., Hastie, T., “Casual interpretations of black-box models”, in *Journal of Business & Economic Statistics*, XXXIX (2021), pp. 272-281, doi:<https://doi.org/10.1080/07350015.2019.1624293>
- [70] Zheng *et al.*, “DRN: A Deep Reinforcement Learning Framework for News Recommendation”, in *WWW 2018: proceedings of the 2018 World Wide Web Conference*, 2018, pp. 167-176, doi:<https://doi.org/10.1145/3178876.3185994>

Sitografia

- [71] <https://www.nrdc.org/stories/flooding-and-climate-change-everything-you-need-know#causes>
- [72] <https://cred.be/sites/default/files/CredCrunch64.pdf>
- [73] <https://cittaclima.it/wp-content/uploads/2020/07/Rapporto-Citt%C3%A0-sempre-pi%C3%B9-calde-Legambiente-2020-alta-1.pdf>
- [74] <https://www.kdnuggets.com/2019/11/understanding-boxplots.html>
- [75] <https://machinelearninggeek.com/outlier-detection-using-isolation-forests/>

- [76] <https://arxiv.org/pdf/1902.00567.pdf>
- [77] <https://en.wikipedia.org/wiki/DBSCAN>
- [78] <https://medium.com/dive-into-ml-ai/faster-implementation-of-mahalanobis-distance-using-tensorflow-42f7aa586bac>
- [79] <https://dinhanhthi.com/decision-tree-regression/>
- [80] <https://bradleyboehmke.github.io/HOML/iml.html#partial-dependence>
- [81] <https://www.ijcaonline.org/archives/volume181/number9/kamande-2018-ijca-917609.pdf>
- [82] <https://towardsdatascience.com/hierarchical-clustering-explained-e59b13846da8>
- [83] <https://towardsdatascience.com/silhouette-method-better-than-elbow-method-to-find-optimal-clusters-378d62ff6891>