Università della Calabria

Corso di Laurea Magistrale in
Data Science per le Strategie Aziendali

Tecniche di clustering e classificazione applicate al biomonitoraggio animale

ANNO ACCADEMICO 2023/2024

Indice

Capitolo 1

Introduzione

1.1 Definizione degli obiettivi

L'analisi condotta in questo elaborato è stata suggerita dalla capacità di alcuni animali di fungere da *bioindicatore*. Tessuti e organi di un bioindicatore animale, infatti, possono essere utilizzati per valutare lo stato di salute dell'ambiente, in funzione delle sostanze che vi si accumulano. I dati utilizzati riguardano esemplari di lucertola campestre (*Podarcis siculus*, Rafinesque-Schmaltz, 1810).
Si ipotizza che il campione contenga esemplari provenienti sia da siti ad alto inquinamento che da siti salubri. L'obiettivo è applicare tecniche di clustering per individuare dei raggruppamenti. L'esperto del dominio di applicazione sarà in grado di identificare, in base alla concentrazione degli elementi in traccia, il grado di inquinamento dell'ambiente da cui gli esemplari dello specifico cluster provengono.
I risultati ottenuti verranno utilizzati per addestrare modelli di classificazione, in modo che questi possano successivamente essere impiegati per la predizione del livello di salubrità di un ecosistema.
I risultati sono stati ottenuti tramite *Weka*.

1.2 Organizzazione dei dati

Il dataset è composto da 2032 righe e 14 attributi:

- *Lunghezza*, lunghezza dell'esemplare, misurati in millimetri;
- *Peso*, massa dell'esemplare, misurata in grammi;
- *Rlm*, rapporto tra lunghezza e massa dell'esemplare, indicatore dello stato di salute dell'animale;

- *Cromo*, concentrazione di cromo, misurata in ppB (parti per miliardo);
- *Manganese*, concentrazione di manganese, misurata in ppB (parti per miliardo);
- *Ferro*, concentrazione di ferro, misurata in ppB (parti per miliardo);
- *Cobalto*, concentrazione di cobalto, misurata in ppB (parti per miliardo);
- *Nichel*, concentrazione di nichel, misurata in ppB (parti per miliardo);
- *Rame*, concentrazione di rame, misurata in ppB (parti per miliardo);
- *Zinco*, concentrazione di zinco, misurata in ppB (parti per miliardo);
- *Selenio*, concentrazione di selenio, misurata in ppB (parti per miliardo);
- *Molibdeno*, concentrazione di molibdeno, misurata in ppB (parti per miliardo);
- *Bario*, concentrazione di bario, misurata in ppB (parti per miliardo);
- *Piombo*, concentrazione di piombo, misurata in ppB (parti per miliardo).

Gli attributi presenti nel dataset sono tutti continui.
La distribuzione dei valori assunti dagli attributi è riportata in figura 1.1.

1.3 Analisi esplorativa dei dati

É noto che la lunghezza e la massa di una lucertola campestre sono legate da una relazione lineare. Infatti, il coefficiente di correlazione tra i due attributi, visibile nella matrice di correlazione riportata in figura 1.2, risulta pari a -0.77. La relazione tra *Lunghezza* e *Peso* è raffigurata nel grafico riportato in figura 1.3
Inoltre, il dataset contiene l'attributo *Rlm*, che è definito come il rapporto tra *Lunghezza* e *Peso* ed è generalmente ritenuto più informativo. Per questi motivi, si è ritenuto opportuno rimuovere gli attributi *Lunghezza* e *Peso* dal dataset.
La matrice di correlazione non evidenzia ulteriori correlazioni rilevanti.

L'attributo *Ferro* assume, per la maggior parte delle istanze, valori elevati. Sono presenti, tuttavia, 337 valori pari a 0. Essendo il ferro uno degli

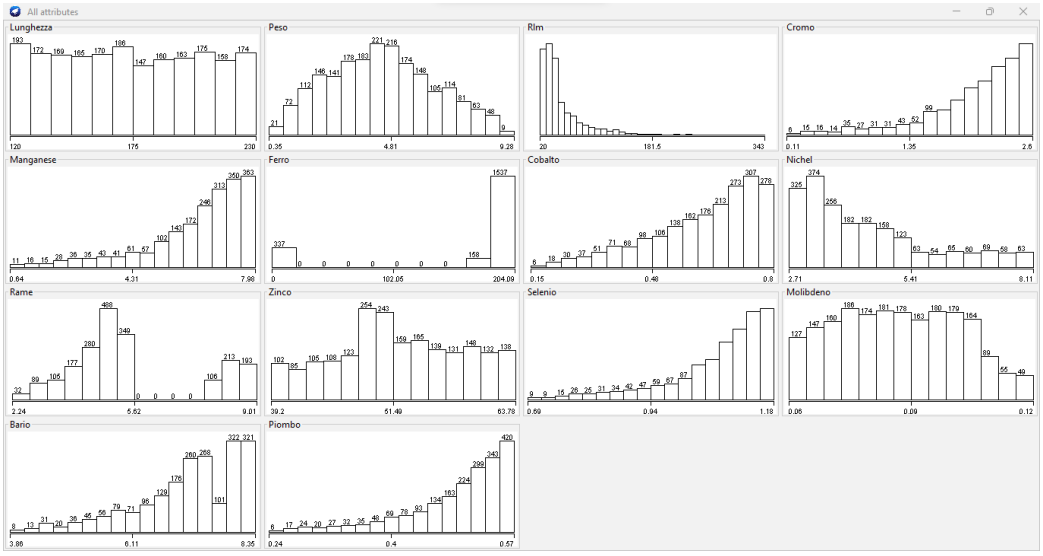

Figura 1.1: Distribuzione dei valori assunti dagli attributi

Correlation matrix

1	0.36	-0.04	0.01	0	0.02	-0.02	0.02	0.02	-0.03	0.01	0.02	-0.02	-0.01
0.36	1	-0.77	-0.26	-0.26	0.3	0.26	-0.45	-0.1	0.5	-0.25	-0.3	-0.14	-0.26
-0.04	-0.77	1	0.22	0.21	-0.22	-0.24	0.39	0.1	-0.42	0.21	0.24	0.12	0.21
0.01	-0.26	0.22	1	0.34	-0.16	-0.24	0.33	0.26	-0.48	0.33	0.44	0.38	0.32
0	-0.26	0.21	0.34	1	-0.17	-0.23	0.34	0.27	-0.48	0.35	0.42	0.4	0.32
0.02	0.3	-0.22	-0.16	-0.17	1	0.18	-0.43	0.13	0.43	-0.16	-0.22	0.02	-0.17
-0.02	0.26	-0.24	-0.24	-0.23	0.18	1	-0.34	-0.07	0.4	-0.23	-0.28	-0.2	-0.24
0.02	-0.45	0.39	0.33	0.34	-0.43	-0.34	1	-0.01	-0.67	0.33	0.44	0.16	0.33
0.02	-0.1	0.1	0.26	0.27	0.13	-0.07	-0.01	1	-0.17	0.26	0.33	0.57	0.27
-0.03	0.5	-0.42	-0.48	-0.48	0.43	0.4	-0.67	-0.17	1	-0.45	-0.59	-0.38	-0.47
0.01	-0.25	0.21	0.33	0.35	-0.16	-0.23	0.33	0.26	-0.45	1	0.39	0.38	0.32
0.02	-0.3	0.24	0.44	0.42	-0.22	-0.28	0.44	0.33	-0.59	0.39	1	0.48	0.43
-0.02	-0.14	0.12	0.38	0.4	0.02	-0.2	0.16	0.57	-0.38	0.38	0.48	1	0.39
-0.01	-0.26	0.21	0.32	0.32	-0.17	-0.24	0.33	0.27	-0.47	0.32	0.43	0.39	1

Figura 1.2: Matrice di correlazione

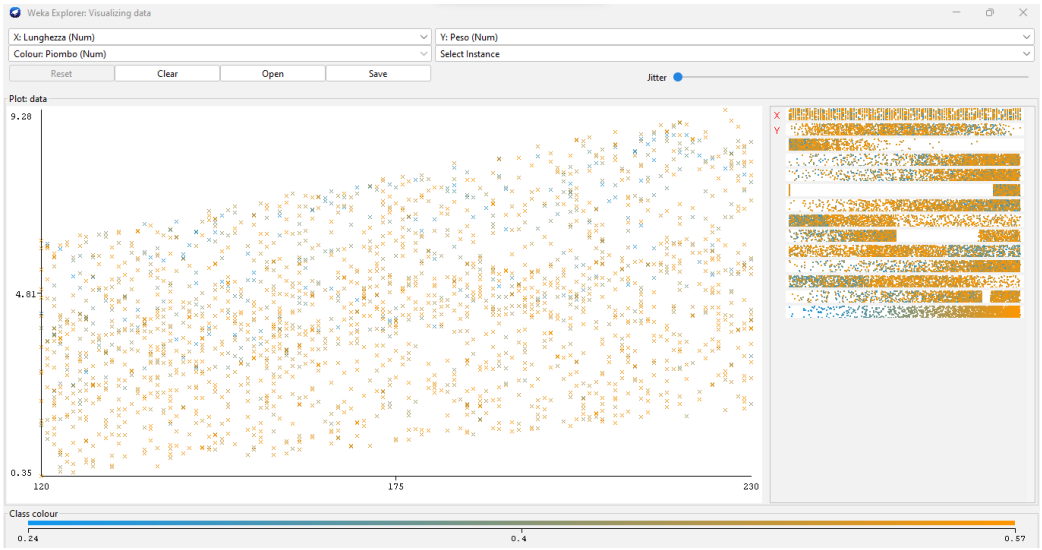

Figura 1.3: Relazione tra *Lunghezza* e *Peso*

elementi maggiormente presenti nella crosta terrestre, i valori nulli non possono che essere attribuiti ad un errore (plausibilmente, il valore 0 è stato utilizzato per indicare valori mancanti). Non avendo, comunque, un ruolo particolarmente discriminante, l'attributo è stato rimosso.

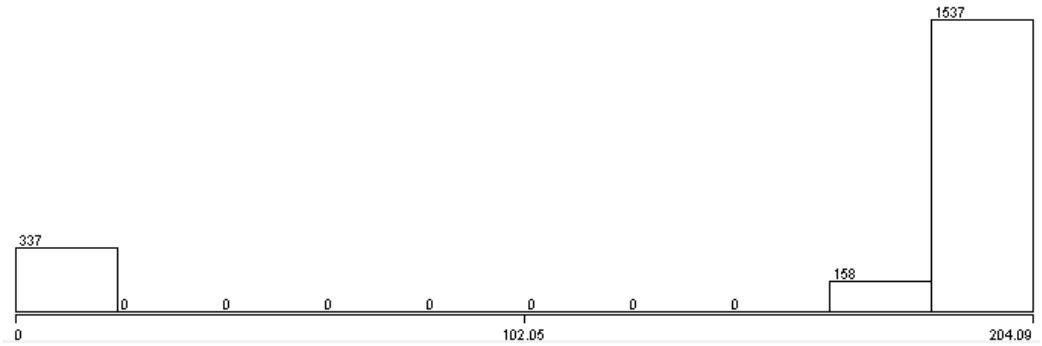

Figura 1.4: Distribuzione dei valori assunti dall'attributo *Ferro*

Il dataset non contiene altri valori mancanti.
Ai fini di migliorare l'interpretabilità dei risultati, gli attributi (tutti continui) sono stati normalizzati.
L'analisi non evidenzia la presenza di outlier.

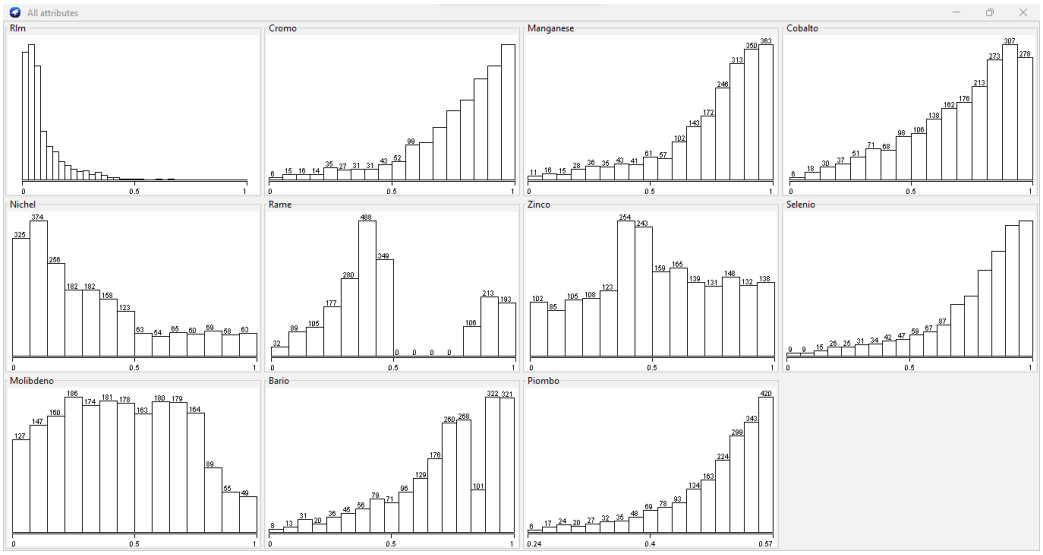

Figura 1.5: Distribuzione dei valori assunti dagli attributi selezionati

Capitolo 2

Clustering

Il clustering è una tecnica utilizzata per organizzare un insieme di oggetti in gruppi, costruiti sulla base della similarità tra essi, in modo tale che

- la similarità intra-cluster sia massimizzata
- la similarità inter-cluster è minimizzata.

Il clustering rientra nell'ambito dell'apprendimento non supervisionato: gli algoritmi vengono addestrati su dati non etichettati e l'obiettivo è comprendere come gli oggetti tendono naturalmente a raggrupparsi.
Nelle sezioni successive, si riportano i risultati ottenuti con l'algoritmo K-means che, in breve, costruisce i cluster assegnando i punti ai cluster ai quali risultano più vicini e aggiornando i centroidi fino alla convergenza. L'utilizzo dell'algoritmo K-means prevede

- l'individuazione di una misura di prossimità
- la scelta del numero di cluster desiderato
- il metodo di inizializzazione dei centroidi.

In questo caso, gli attributi sono tutti continui; dunque, è possibile servirsi della distanza euclidea (già opzione di default in Weka).
Per quanto riguarda, invece, la scelta di K, va specificato che l'efficienza dell'algoritmo permette di effettuare svariate prove valutando le performance al variare del parametro. Tuttavia, in questo caso specifico, l'obiettivo è valutare lo stato di salubrità di un ecosistema e si ipotizza la naturale esistenza di zone totalmente naturali, totalmente antropizzate (e, quindi, soggette a un maggiore inquinamento) e zone di livello intermedio, in cui l'antropizzazione non è tale da causare un elevato tasso di inquinamento. Per questi motivi, il

numero di cluster è stato impostato a 3.
I risultati dell'algoritmo sono influenzati dal modo in cui si individuano i centroidi. In questo elaborato, sono stati utilizzati due metodi: random e K++.

2.1 K-means con inizializzazione random

In questo caso, i centroidi vengono casualmente scelti tra i punti forniti in input. I risultati sono riportati in Figura 2.1 e Figura 2.2.

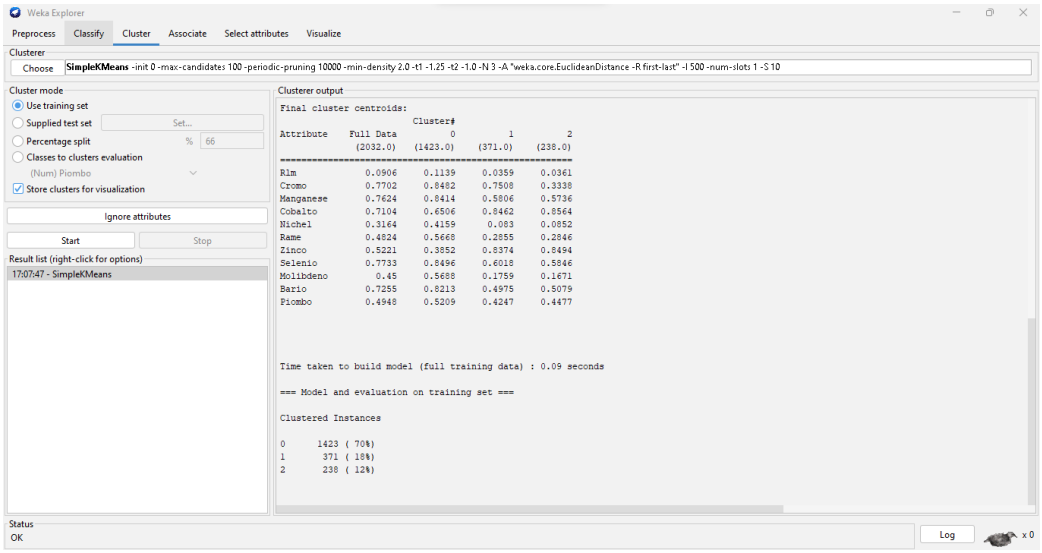

Figura 2.1: Output dell'algoritmo K-means con inizializzazione random

```
Number of iterations: 17
Within cluster sum of squared errors: 680.6416880436742

Initial starting points (random):

Cluster 0: 0.120743,0.784512,0.616305,0.677862,0.330236,0.442109,0.371768,0.594615,0.697865,0.751202,0.566027
Cluster 1: 0.024768,0.931372,0.833717,0.83738,0.137325,0.40164,0.813506,0.549711,0.167488,0.300086,0.34349
Cluster 2: 0.018576,0.196721,0.99248,0.992727,0.040389,0.098481,0.910366,0.869723,0.239737,0.215871,0.292572
```

Figura 2.2: Errore dell'algoritmo K-means con inizializzazione random

2.2 K-means con inizializzazione K++

Nel metodo K++, il primo centroide viene scelto casualmente; tutti gli altri vengono selezionati in modo da essere il più lontano possibile dagli altri.

I risultati ottenuti sono riportati in Figura 2.3 e Figura 2.4.

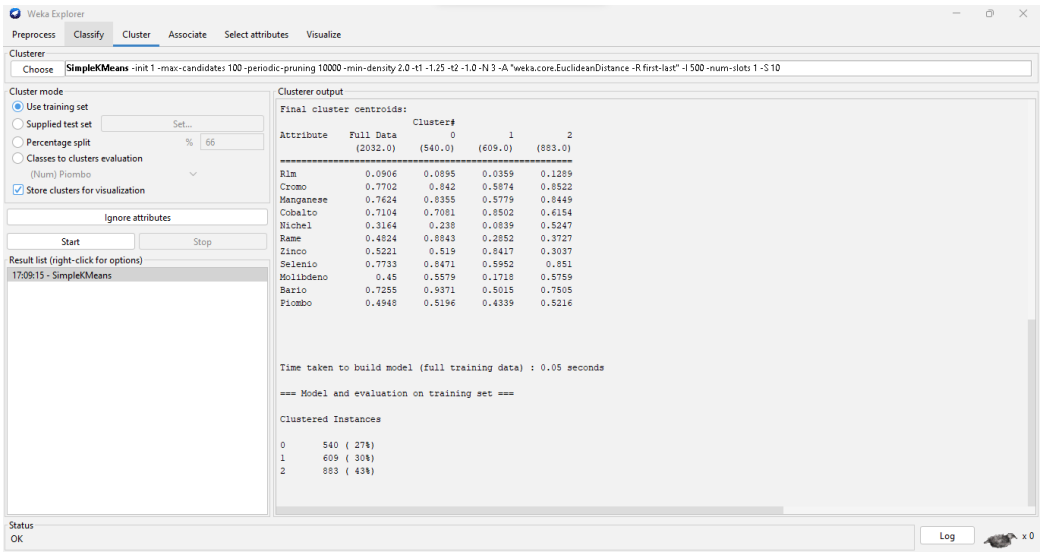

Figura 2.3: Output dell'algoritmo K-means con inizializzazione k++

```
Number of iterations: 12
Within cluster sum of squared errors: 560.6034198720743

Initial starting points (k-means++):

Cluster 0: 0.120743,0.784512,0.616305,0.677862,0.330236,0.442109,0.371768,0.594615,0.697865,0.751202,0.566027
Cluster 1: 0.024768,0.096796,0.205125,0.920762,0.014771,0.30908,0.77162,0.30234,0.272578,0.7169,0.537956
Cluster 2: 0.027864,0.773818,0.478186,0.876932,0.534179,0.335229,0.068044,0.837667,0.770115,0.746644,0.465695
```

Figura 2.4: Errore dell'algoritmo K-means con inizializzazione k++

I risultati ottenuti modificando il metodo di inizializzazione dei centroidi sono sostanzialmente diversi. In particolare, i cluster risultano avere dimensioni nettamente diverse. Si ritiene che l'utilizzo del metodo K++ permetta di ottenere risultati che meglio rispecchiano la struttura intrinseca dei dati, rispetto a quanto si ottiene con un'inizializzazione completamente casuale. Per questo motivo, i cluster determinati con il metodo K++ saranno quelli utilizzati nel capitolo successivo.
La distribuzione dei valori assunti dagli attributi ripartiti per cluster di appartenenza è riportato in Figura 2.5. L'assegnazione del livello di salubrità dell'ambiente in funzione delle concentrazioni di elementi in traccia sarà compito dell'esperto del dominio di applicazione.

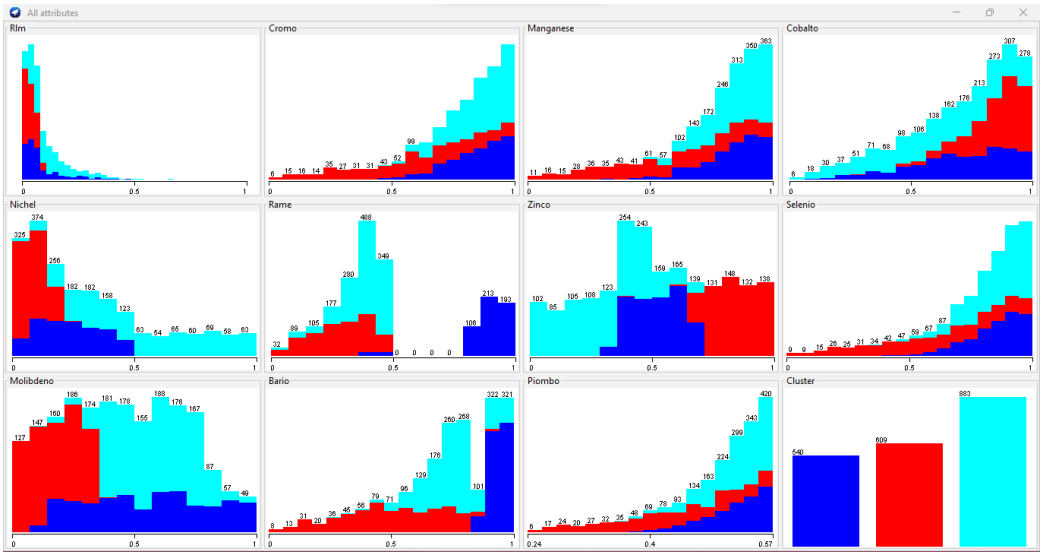

Figura 2.5: Distribuzione dei valori assunti dagli attributi ripartiti per cluster di appartenenza

Capitolo 3

Classificazione

I risultati ottenuti sono stati esportati in un file .arff e utilizzati come input per svariati algoritmi di classificazione.
I problemi di classificazione rientrano nell'apprendimento supervisionato. In questo ambito, è necessaria l'individuazione di una variabile target e che i dati raccolti siano etichettati rispetto ad essa. Il modello ottenuto in fase di addestramento può essere, in seguito, utilizzato per predire la classe di appartenenza di un nuovo record.
In questo caso, l'attributo di classe è *Cluster* (rinominato, per una maggiore leggibilità, come *Zona*).
Per valutare le performance di un modello di classificazione, sono necessari due insiemi di oggetti, disgiunti e indipendenti:

- il *training set*, utilizzato per indurre il modello
- il *test set*, una collezione complementare composta da oggetti della stessa forma e della stessa origine di quelli contenuti nel training set. Il test set viene utilizzato per validare l'accuratezza del modello, rispetto a una serie di misure, confrontando la predizione con l'etichetta di cluster originaria.

Una delle misura che consente di quantificare la validità di un modello è l'accuratezza, definita come la frazione di istanze del test set di cui sono state riconosciute correttamente le etichette.
Per ognuno dei classificatori impiegati nelle successive sezioni, sono state utilizzate due tecniche:

- un'unica ripartizione dell'intero dataset in training set e test set con una percentuale del 66
- la k-fold cross validation, con k = 10.

La seconda soluzione prevede di suddividere il dataset in k sottoinsiemi. Il modello viene addestrato k volte: a ogni iterazione k-1 sottoinsiemi vengono utilizzati come training set e il sottoinsieme restante come insieme di valutazione. Alla fine del processo, il modello viene valutato in funzione della media delle prestazioni dei k modelli.
La k-fold cross validation ha il vantaggio di utilizzare l'intero dataset sia come training set che come test set, e quindi condurre a risultati più affidabili. Weka, inoltre, applica di default la *stratified cross validation*, cioè si assicura che la distribuzione delle classi originariamente presente nell'intero dataset si mantenga in ognuno dei sottoinsiemi.

3.1 ZeroR

L'algoritmo ZeroR è un algoritmo di classificazione molto semplice: sostanzialmente, non utilizza nessuna informazione proveniente dagli attributi e si limita ad assegnare ogni istanza alla classe più frequente. Chiaramente, non è in grado di fornire predizioni accurate, ma può essere utilizzato per valutare le prestazioni di altri classificatori: infatti, se un modello mostra risultati peggiori di quelli ottenuti tramite ZeroR, allora è indispensabile rivalutare l'intero approccio.

3.1.1 ZeroR con split in training e test set

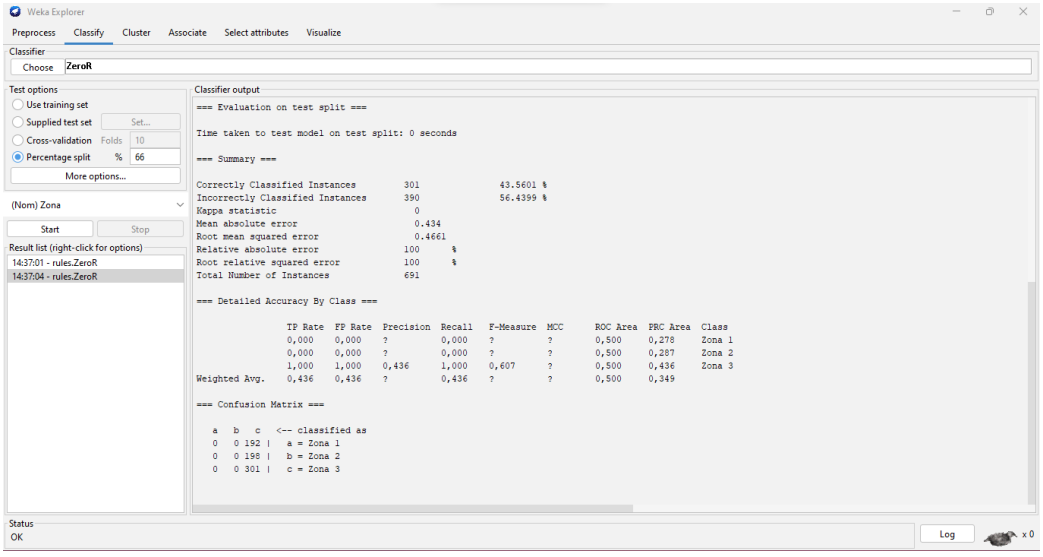

Figura 3.1: Output dell'algoritmo ZeroR con split in training e test set

3.1.2 ZeroR con k-fold

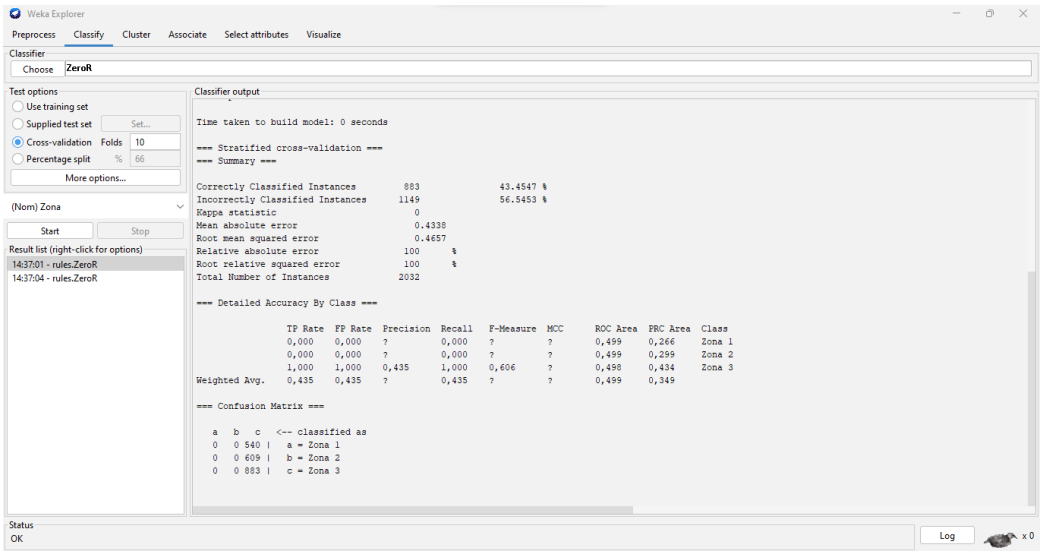

Figura 3.2: Output dell'algoritmo ZeroR con metodo k-fold

3.2 Naive Bayes

Il classificatore Naive Bayes è un modello di classificazione di tipo probabilistico. Prevede di calcolare, con l'ausilio del teorema di Bayes, la probabilità che un'istanza query appartenga a ogni classe, e poi, semplicemente, scegliere quella più probabile.
É necessaria, tuttavia, l'assunzione di indipendenza tra gli attributi che è, in linea teorica, molto restrittiva. In ogni caso, l'esperienza riporta un elevato numero di casi di successo.

3.2.1 Naive Bayes con split in training e test set

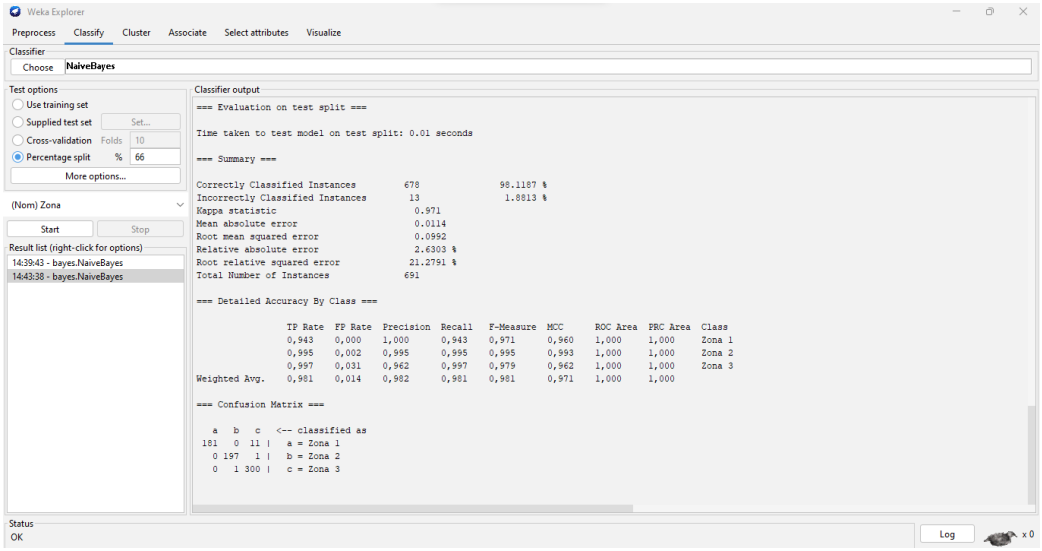

Figura 3.3: Output dell'algoritmo Naive Bayes con split in training e test set

3.2.2 Naive Bayes con k-fold

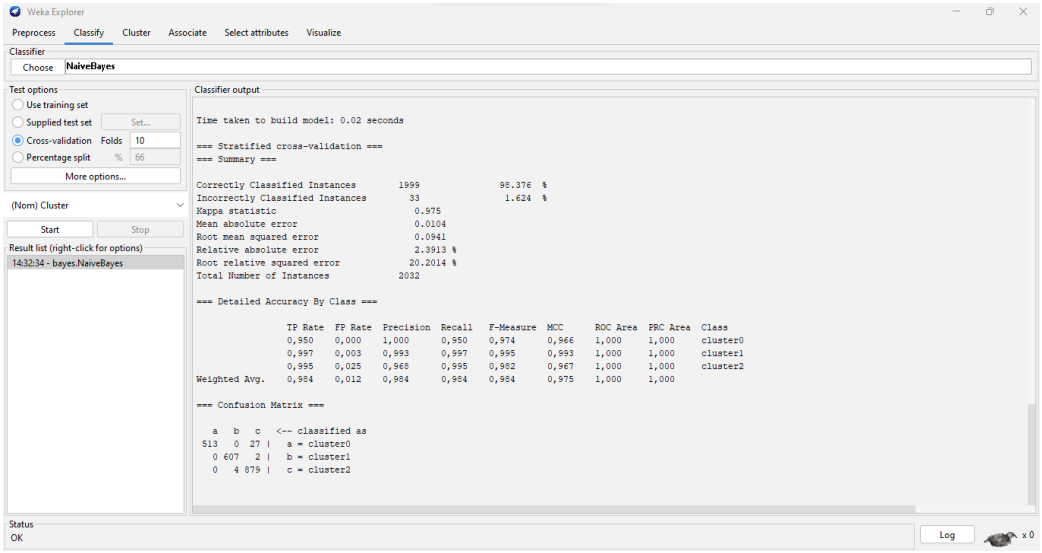

Figura 3.4: Output dell'algoritmo Naive Bayes con metodo k-fold

3.3 J48

L'algoritmo J48 è l'implementazione di C4.5 nell'ambiente Weka.
Il criterio per individuare la migliore ripartizione fa leva sull'entropia, valutata su una distribuzione delle istanze rispetto alle etichette. L'obiettivo è massimizzare la riduzione di entropia, e quindi individuare l'attributo che porta a un'entropia che sia la più bassa possibile. La misura della riduzione dell'entropia viene chiamata *information gain*.
Il classificatore J48 è robusto e fornisce in output un albero di decisione, che si traduce in regole IF THEN ELSE, per cui gode di un'elevatissima interpretabilità. Tuttavia, un albero di decisione troppo profondo e troppo specifico perde di espressività; quindi, al fine di limitare la super specializzazione del modello, il numero minimo di oggetti in ogni nodo fogli è stato impostato a 20.

3.3.1 J48 con split in training e test set

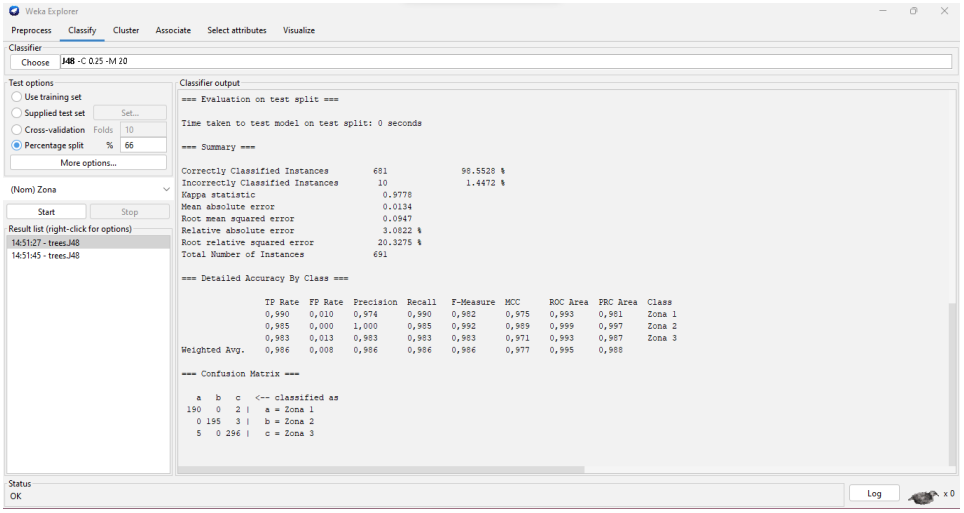

Figura 3.5: Output dell'algoritmo J48 con split in training e test set

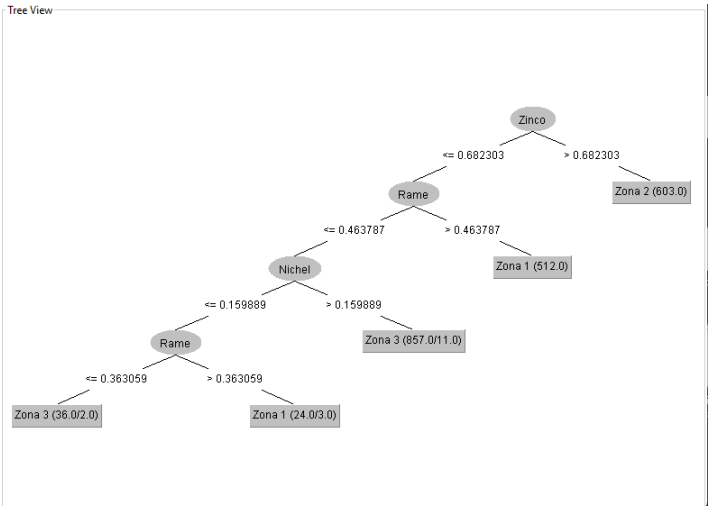

Figura 3.6: Albero di decisione ottenuto con l'algoritmo J48 con split in training e test set

3.3.2 J48 con k-fold

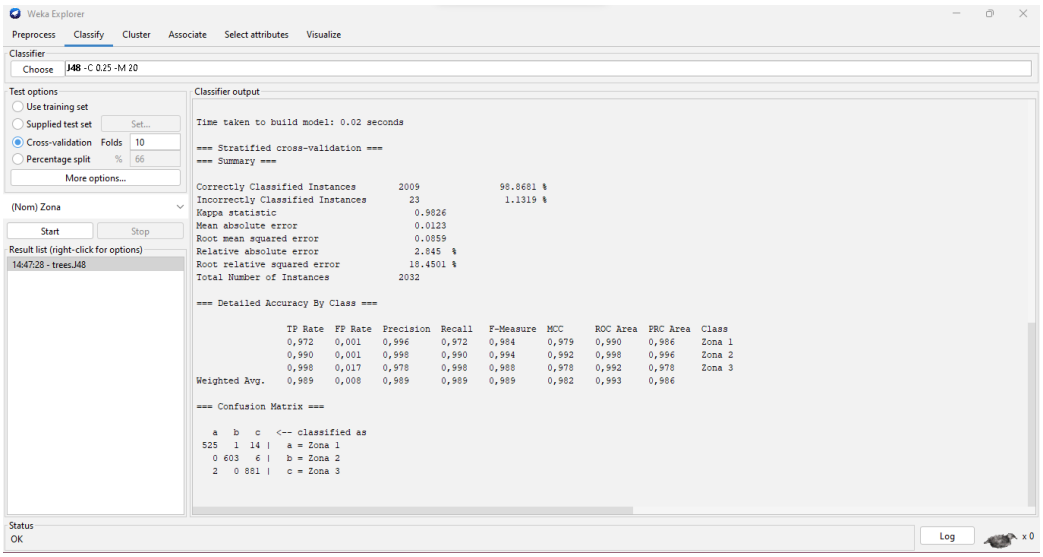

Figura 3.7: Output dell'algoritmo J48 con metodo k-fold

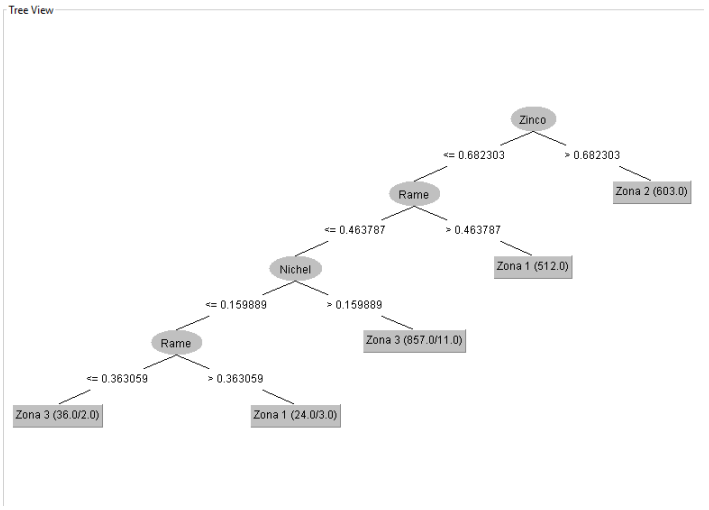

Figura 3.8: Albero di decisione ottenuto con l'algoritmo J48 e metodo k-fold

3.4 IBk

L'algoritmo IBk (Instance-Based k-Nearest Neighbors) è una versione dell'algoritmo k-Nearest Neighbors. Per classificare una nuova istanza, determina la classe più comune tra i k vicini più vicini, individuati calcolando la prossimità dell'istanza query da ogni altra.
Variare il parametro k consente di rendere il modello più robusto alle variazioni casuali nel dataset.

3.4.1 IBk con split in training e test set e k = 1

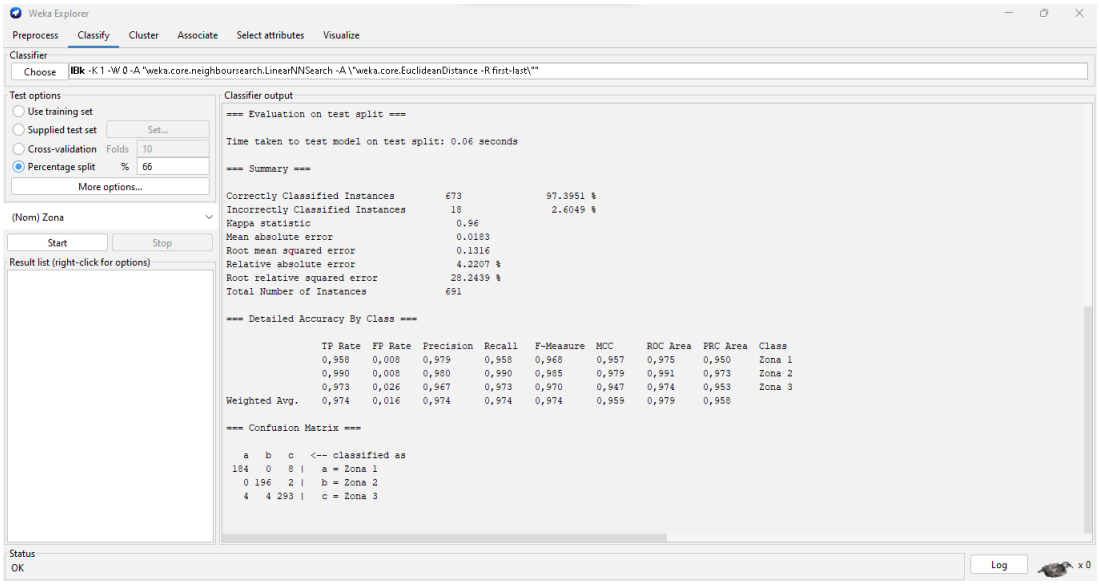

Figura 3.9: Output dell'algoritmo IBk con split in training e test set e k = 1

3.4.2 IBk con k-fold e k = 1

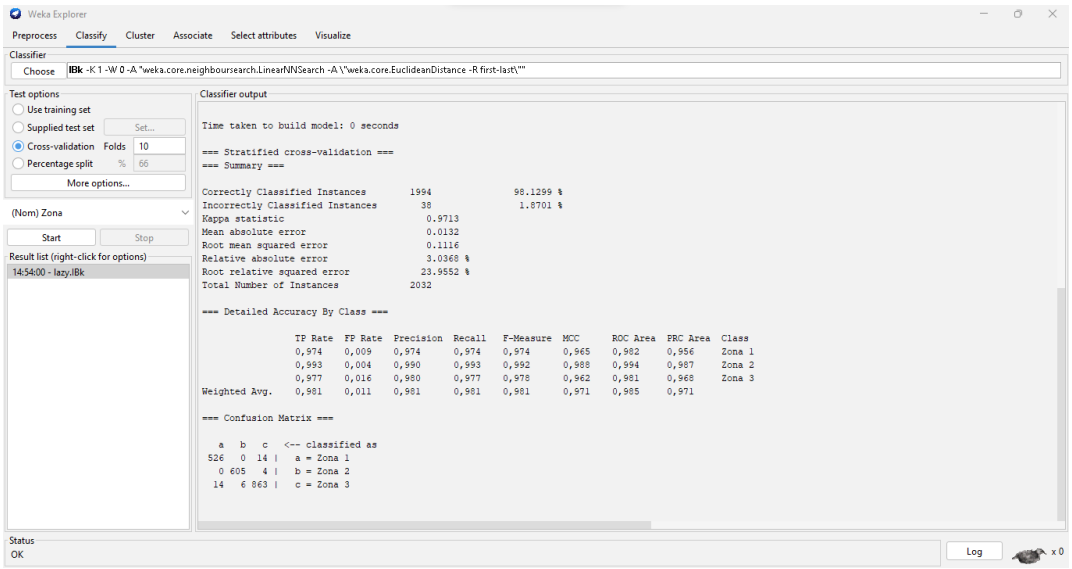

Figura 3.10: Output dell'algoritmo IBk con metodo k-fold e k = 1

3.4.3 IBk con split in training e test set e k = 5

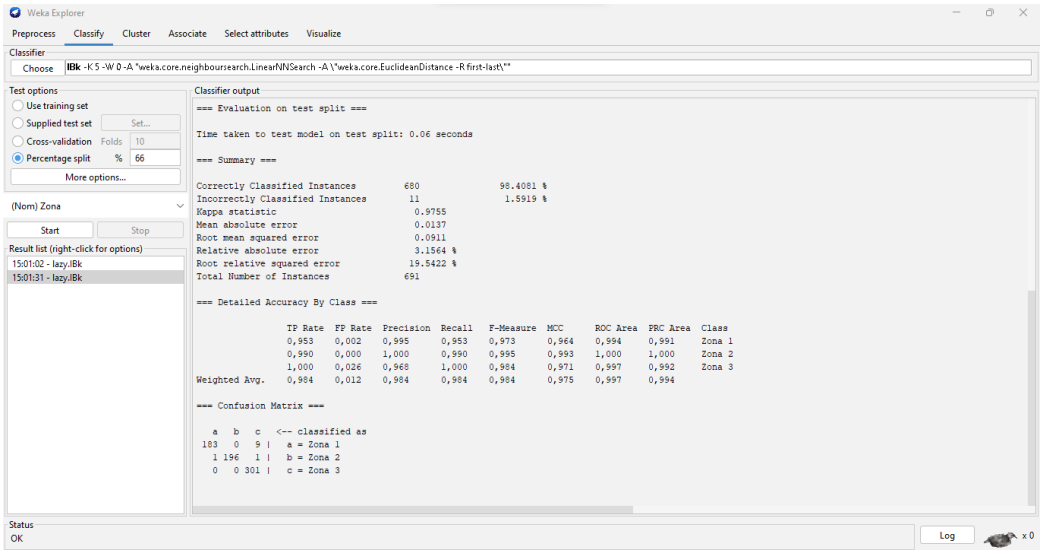

Figura 3.11: Output dell'algoritmo IBk con split in training e test set e k = 5

3.4.4 IBk con k-fold e k = 5

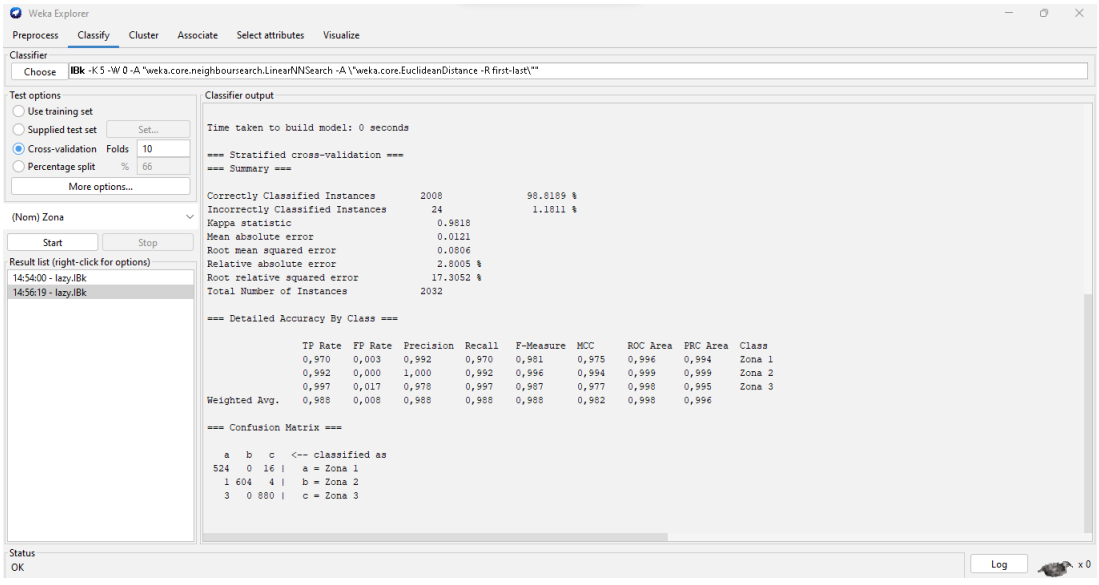

Figura 3.12: Output dell'algoritmo IBk con metodo k-fold e k = 5

3.5 Random Forest

L'algoritmo Random Forest combina i risultati di più alberi decisionali, ognuno dei quali è addestrato su un sottoinsieme delle istanze e dei dati. La predizione finale del modello è determinata dalla maggioranza.

3.5.1 Random Forest con split in training e test set

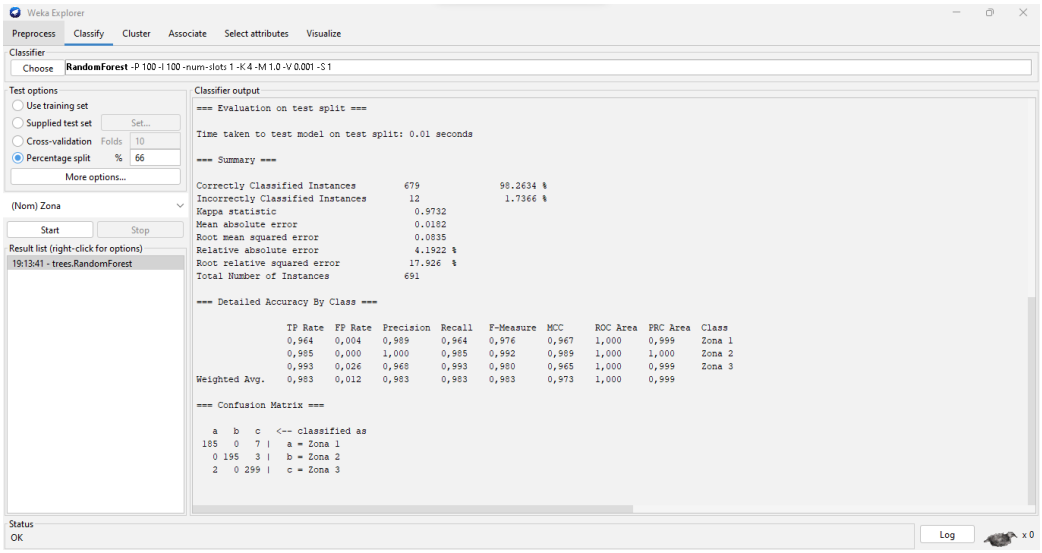

Figura 3.13: Output dell'algoritmo Random Forest con split in training e test set

3.5.2 Random Forest con k-fold

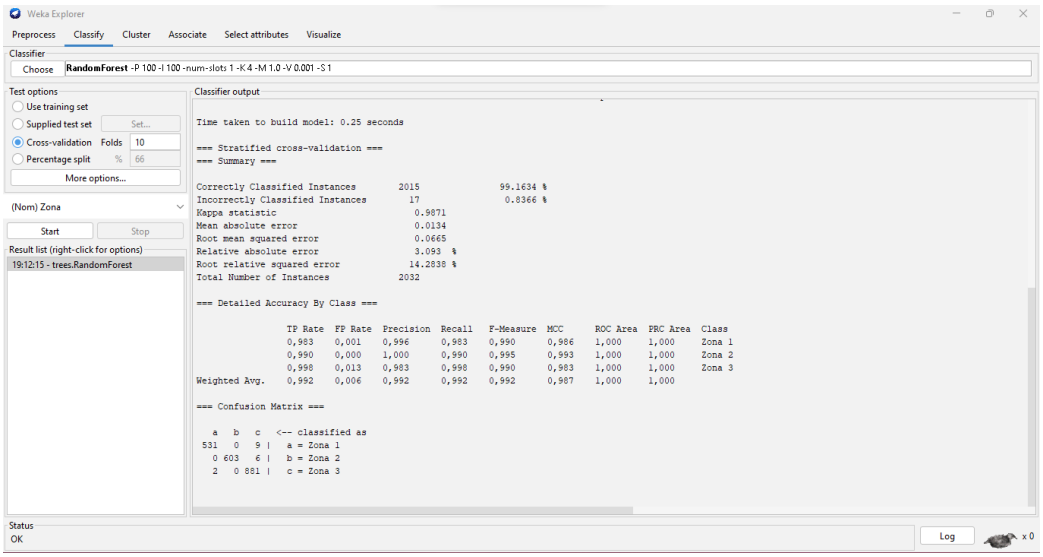

Figura 3.14: Output dell'algoritmo Random Forest con metodo k-fold

Capitolo 4

Conclusioni

I risultati ottenuti sono riassunti nella seguente tabella.

Classificatore	Metodo	Accuratezza
ZeroR	Split unico	43.56
ZeroR	K-fold	43.45
Naive Bayes 3	Split unico	98.12
Naive Bayes	K-fold	98.38
J48	Split unico	98.55
J48	K-fold	98.87
IBk	Split unico, k = 1	97.31
IBk	K-fold, k = 1	98.13
IBk	Split unico, k = 5	98.41
IBk	K-fold, k = 5	98.82
Random Forest	Split unico	98.26
Random Forest	K-fold	99.16

L'accuratezza maggiore è raggiunta con l'algoritmo Random Forest. In ogni caso, tutti gli algoritmi (tranne, chiaramente, lo ZeroR) restituiscono ottimi livelli di accuratezza.