Assegnamento 1 : Decision Trees

January 24, 2022

Nicolò Toscani

1 Esercizio 1

1.1 Esercizio 1a

Problema: dato un punto con coordinate x e y stabilire se appartiene ad un cerchio oppure ad un quadrato.

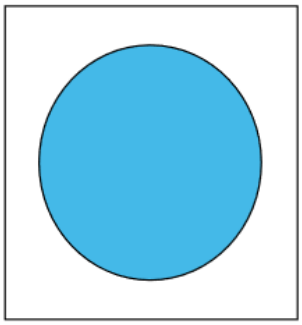

Figure 1: Regione di interesse

Viene caricato il training set *circletrain.arff* che contiene 53 punti contenuti all'interno del cerchio centrato in 0,0 con raggio 1 e 47 punti contenuti nel quadrato esterno al cerchio con un lato di lunghezza L = 2.5.
Ogni istanza del training set è composta dalle coordinate x, y e una corrispettiva label di appartenza.

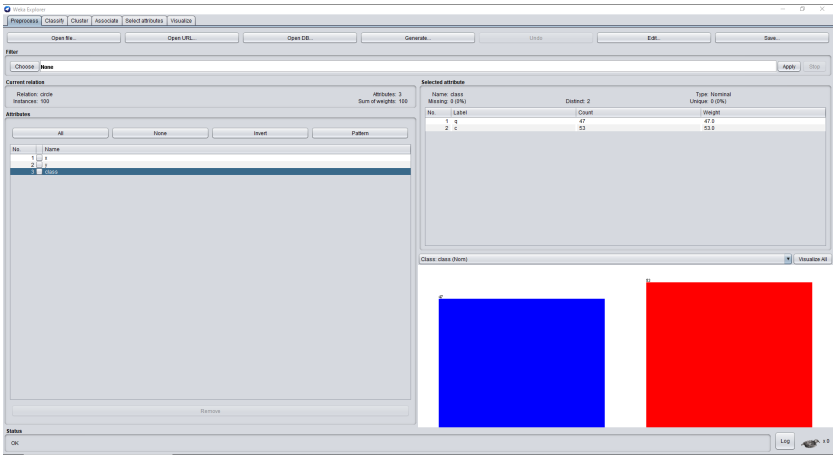

Figure 2: Training set

Viene selezionato in *classificatore* da utilizzare. Per questo esercizio viene utilizzato il classificatore **IBk** (*K-Nearest Neighbours*) la cui accuratezza viene valutata sul test set fornito *circletest.arff*.
L'obbiettivo é quindi quello di classificare i punti che vengono forniti nel test set *circletest.arff* utilizzando l'agoritmo *KNN* addestrato sul training set *circletrain.arff*.

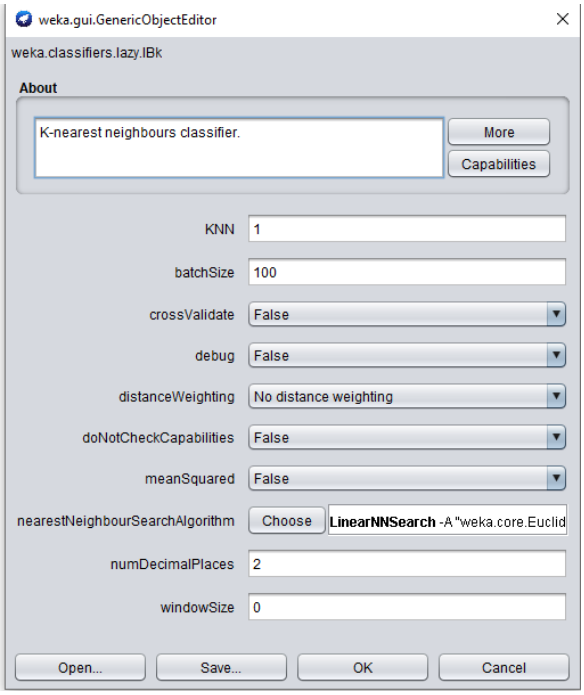

Figure 3: Classificatore KNN

Un oggetto é classificato in base alla maggioranza dei voti dei suoi k vicini. L'accuratezza del classificatore viene valutata al variare del parametro k (numero dei vicini da utilizzare). Per evitare situazioni di paritá vengono scelti dei parametri dispari.

k	Accuratezza
1	91%
3	88%
5	87%
7	84%

Table 1: Valori accuratezza *circletest.arff*

Al variare del parametro k otteniamo un decremento del valore di accuratezza.
Il valore di accuratezza maggiore lo si ottiene ponendo $k = 1$ dove per assegnare un nuovo punto estratto dal *test set* si prende come riferimento solamente un *unico* punto piú vicino contenuto nel *training set*.

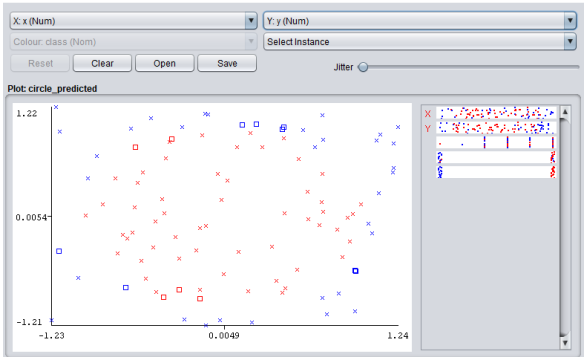

Figure 4: *Errori di classificazione sul test set*

k	Accuratezza
1	89.96%
3	89.00%
5	86.88%
7	86.04%

Table 2: Valori accuratezza *circleall.arff*

Successivamente vengono confrontati i risultati di accuratezza con un *test set* che contiene istanze di valori piú densi (2600 campioni) e uniformi sul dominio in esame. Il file *circleall.arff* viene utilizzato quindi come *test set*. Anche con questo *test set* possiamo notare che la classificazione di appartenenza alla relativa classe *quadrato / cerchio* ottiene risultati migliori con il parametro $k = 1$.

1.2 Esercizio 1b

Problema: sostituire la regione interna (cerchio) con un quadrato di lato L = 1.76. Entramhbi i quadrati sono quindi centrati nel punto (0,0) del piano in esame.

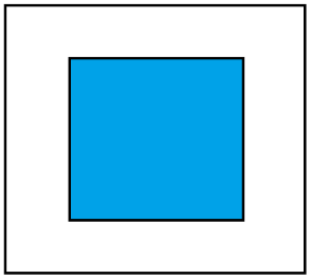

Figure 5: Nuova regione

Vengono modificati i dati contenuti nel *training set* e nel *test set* per soddisfare le nuove specifiche richieste.
Le classi di appartenenza vengono suddivise in:

- Q: quadrato di lato L = 2.5
- q: quadrato di lato L = 1.76

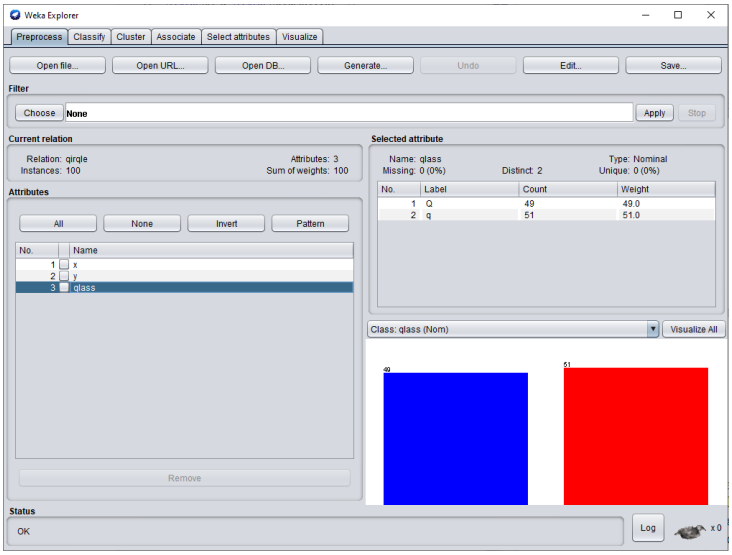

Figure 6: Training set

Utilizzando il classificatore *J48* (implementazione del C4.5) viene prodotto un albero di decisione facilmente traducibile in *IF-ELSE* in grado di classificare i dati osservando in pattern presenti nel *training set*.

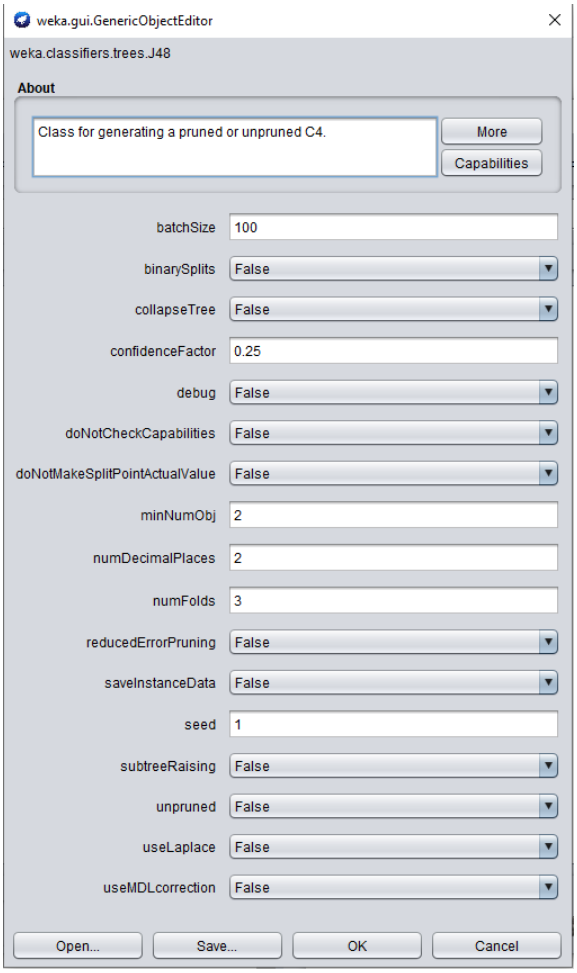

Figure 7: Classificatore J48

L'accuratezza ottenuta utilizzando questo tipo di classificatore é del 93%. Risulta quindi essere migliore rispetto a quella ottenuta utilizzando l'algoritmo *K-NN*. Il problema in esame risulta essere più adatto a problemi di decisione basati sulle coordinate del punto fornito in ingresso invece di utilizzare una classificazione basata sulla distanza tra il vicinato come *K-NN*.

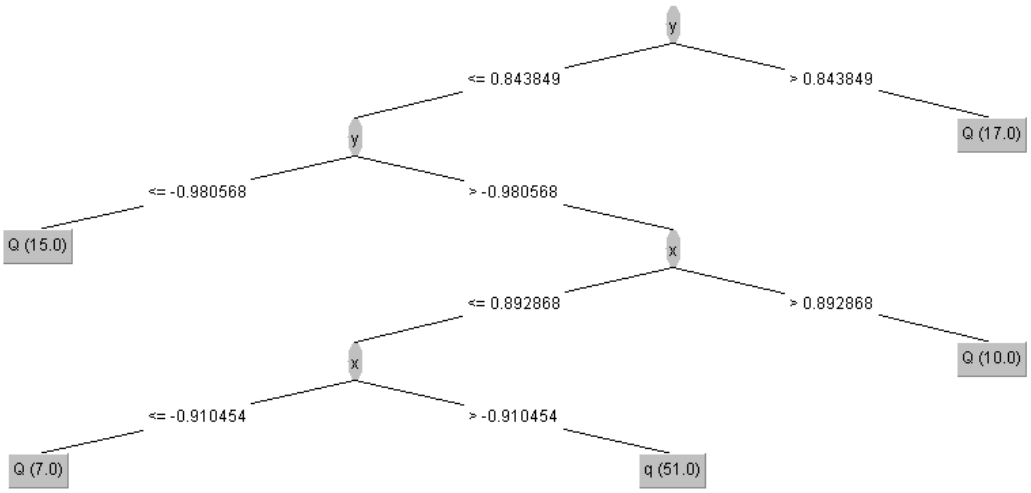

Figure 8: Decision tree (x,y)

Come é facile notare, l'albero puó essere facilmente tradotto in istruzioni condizionali *IF-ELSE* basate sulle coordinate dei punti (x,y) per essere classificati nella corretta classe di appartenenza.

Dati:

- L1 = 1.76 (lato del quadrato piccolo q)
- L2 = 2.5 (lato del quadrato grande Q)

possiamo definire un equazione di appartenenza dei punti alle due distinte regioni di interesse:

$$(x, y) \in q$$

se

$$x \leq \frac{L1}{2} \wedge x \geq -\frac{L1}{2} \wedge y \leq \frac{L1}{2} \wedge y \geq -\frac{L1}{2}$$

altrimenti

$$(x, y) \in Q$$

Successivamente sul *dataset* originale modifichiamo la rappresentazione dei dati in:

$$z = x^2, t = y^2$$

Quello che ci si aspetta di ottenere é un accuratezza come quella della rappresentazione originale in quanto l'unica modifica che puó essere apportata é il cambio di segno di una coordinata con relativo cambio di proporzionalitá. L'accuratezza che si ottiene é dell' 88%.

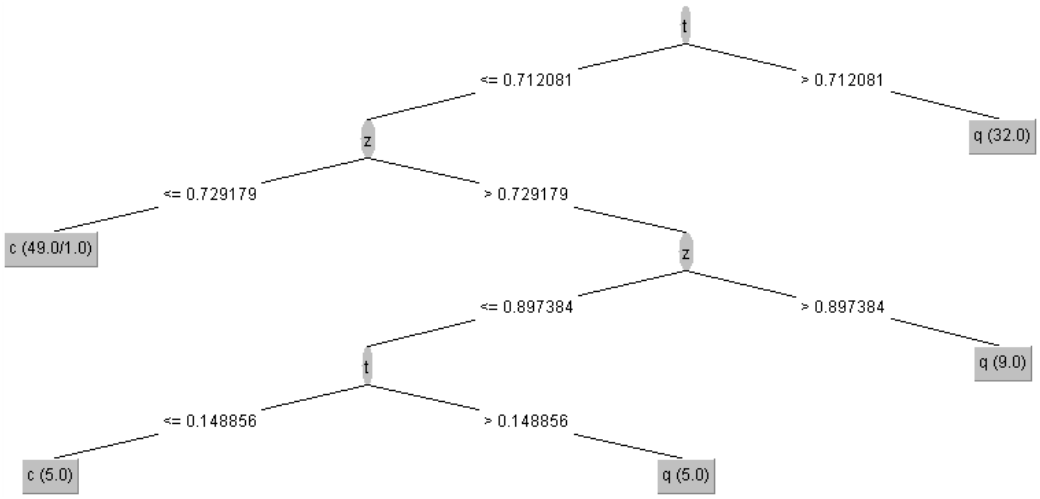

Figure 9: Decision tree (z,t)

É inoltre possibile definire una nuova variabile funzione di x e y per ottenere un accuratezza del 100 % con il minore albero possibile.
Possiamo scegliere la seguente rappresentazione:

$$d = \sqrt{x^2 + y^2}$$

Riformulando il *training set* e il *test set* secondo questa rappresentazione otteniamo un accuratezza del 100%.

$$(x, y) \in c$$

se

$$d \leq 1$$

altrimenti

$$(x, y) \in q$$

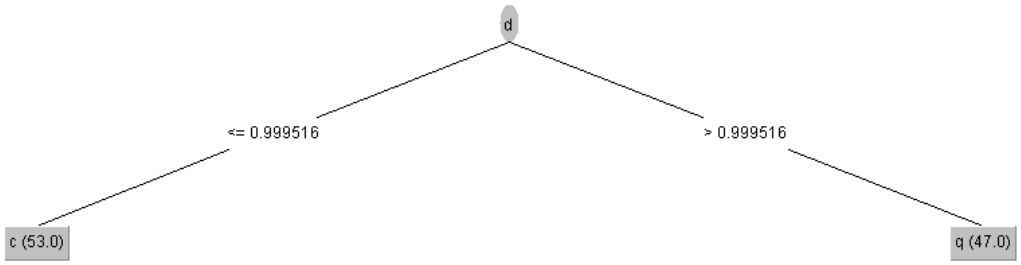

Figure 10: Decision tree (d)

É possibile notare che unificando i valori della x e y in un unica *feature* é possibile ottenere un migliore risultato con un albero di mensione ridotta formato da una sola condizione di *IF-ELSE*. Riformulando anche la rappresentazione di *circleall.arff* si ottiene un accuratezza del 99.7691%.

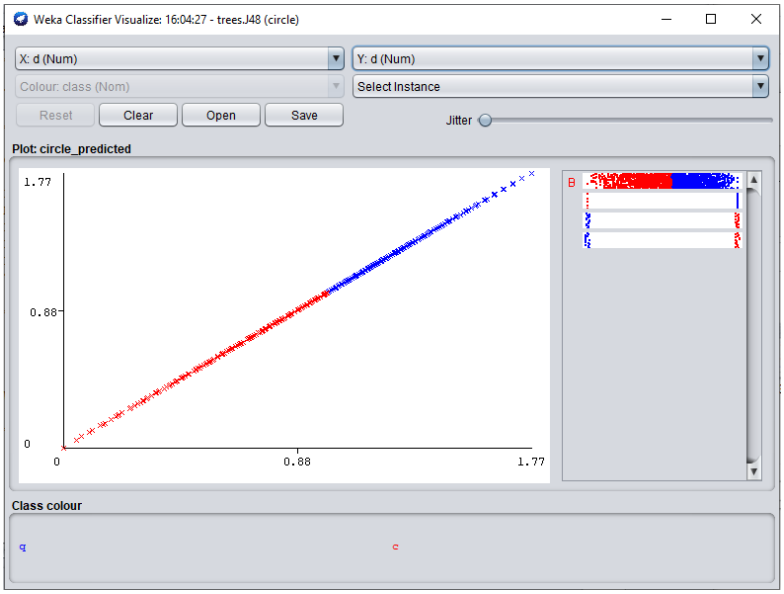

Figure 11: Classifier errors (d)

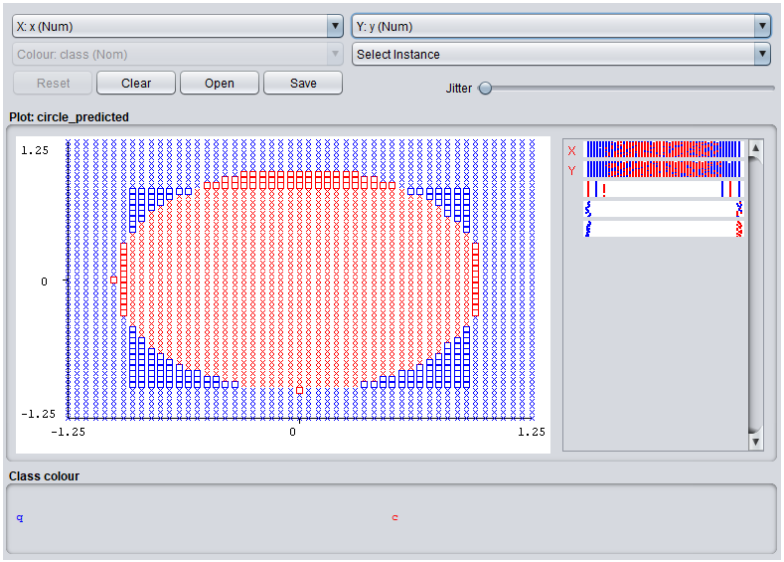

Figure 12: Classifier errors (x,y)

2 Esercizio 2

2.1 Esercizio 2a

Dopo aver caricato il file $Bigtest1\_104.arff$ che contiene la rappresentazione binaria delle cifre da 0 a 9 delle cifre delle targhe si procede alla classificazione delle etichette. L'obbiettivo é quello di riuscire ad assegnare la codifica binaria del carattere digitale alla corretta cifra.
L'immagine in arrivo viene rappresentata come un array di 104 valori binari. Utilizzando come *test set* uno split random del 66% si ottiene un accuratezza del 96.7773% con il seguente albero di decisione:

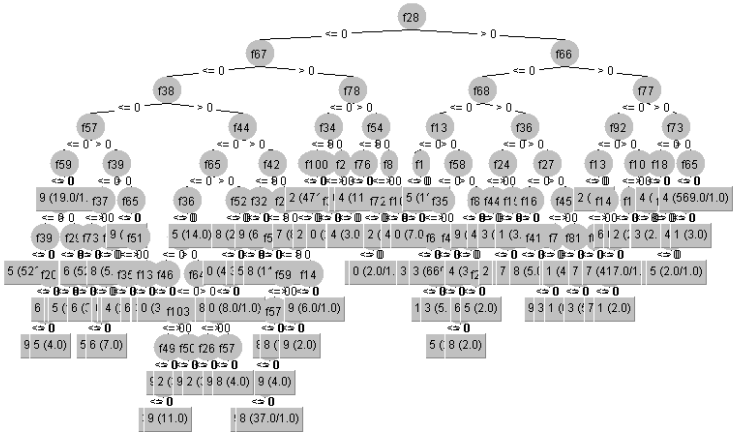

Figure 13: Decision tree

Per aumentare il grado di generalitá del risultato l'algoritmo viene eseguito 5 volte modificando il *seed*.

seed	Accuratezza
1	96.773%
2	96.0449%
3	96.0938%
5	95.8008%
100	96.875%

Table 3: Accuratezza variando seed

L'accuratezza media risulta essere di 96.3175%.
Utilizzando il file $Bigtest2\_104.arff$ come *test set* si ottiene un accuratezza del 95.3094%.

2.2 Esercizio 2b

Utilizzando come *training set* il file $Bigtest1\_104.arff$ e come *test set* il file $Bigtest2\_104.arff$ viene valutata l'accuratezza dell'algoritmo *J48* mantenendo tutti i valori di default ma variando il parametro M (minimo numero di campioni del trainig set classificati in ciascuna foglia dell'albero di decisione).
Con M che tende ad 1 otteniamo un numero maggiore di diramazioni sull'albero decisionale ottenendo un modello piú specifico sul problema in esame.

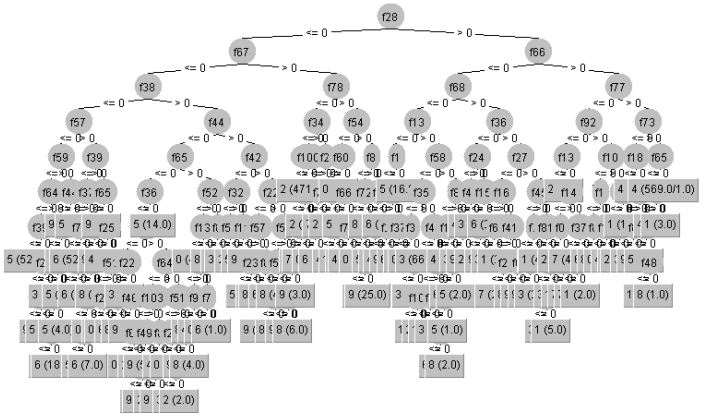

Figure 14: Decision tree con M = 1

M	Accuratezza
1	95.1497%
2	95.3094%
3	95.3493%
4	95.509%
5	95.2295%
10	94.4112%
50	89.022%
100	87.0858%

Table 4: Accuratezza variando M

M	Accuratezza
1	99.751%
2	99.1036%
3	98.5558%
4	98.2902%
5	97.8918%
10	97.1448%
50	92.3307%
100	89.890%

Table 5: Accuratezza variando M sul training set

Successivamente i risultati ottenuti vengono confrontati con i valori che si ottengono utilizzando come *test set* il *training set*.

Partendo da un valore di M inadeguato (troppo grande) é possibile decrementarlo in modo da migliorare la capacitá del classificatore di adattarsi e distinguere correttamente le diverse classi.
Partendo dal valore 100 e decrementando l'accuratezza aumenta fino a che non si verifica *overfitting*. A partire dal valore di M uguale a 3 sul *test set* l'accuratezza inizia a decrementare mentre continua a salire sul modello testato sul *training set*.
Il valore **ottimo** per il parametro M é 4.