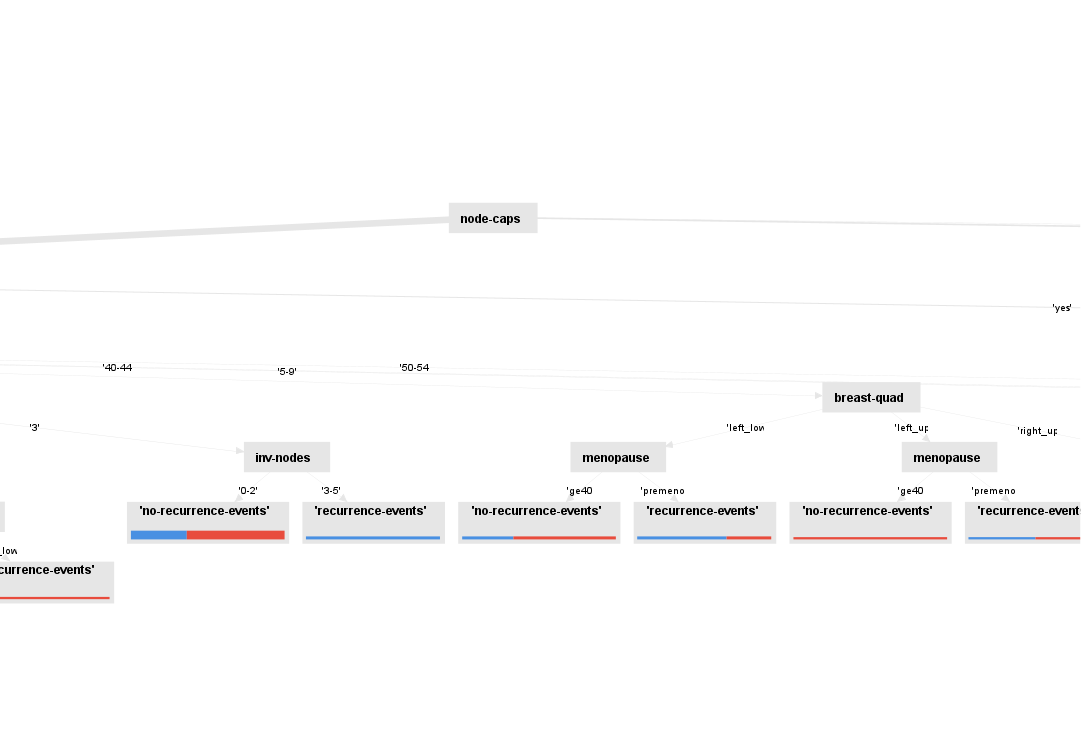
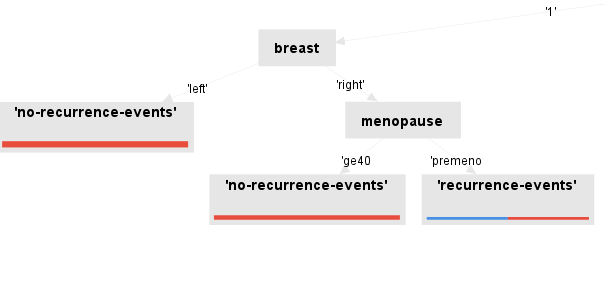
**Domanda 1**

1. L’attributo più selettivo è la radice del nostro albero di decisione, quindi nel nostro caso “node-caps”. 
2. L’altezza dell’albero è 6. L’altezza è la lunghezza massima di un percorso che collega una foglia alla radice.
3. Un partizionamento puro è uno split sui valori di un attributo che fa in modo che i record corrispondenti appartengano tutti alla stessa classe. Nella figura sottostante si può notare come la partizione di sinistra, ottenuta da split su valori dell’attributo “breast”, sia pura in quanto tutte le relative istanze fanno parte della stessa classe “no-recurrence-events”.



**Domanda 2**

Il parametro “minimal-gain” permette di scegliere il threshold per cui scegliere se continuare a splittare oppure no. Un nodo viene splittato se il suo gain è superiore al valore di “minimal-gain”. Se il valore di minimal gain è alto gli alberi saranno più piccoli, mentre valori bassi permettono di creare alberi più fitti e alti. Valori molto alti possono fare in modo che l’albero abbia un solo nodo, come vediamo nella Figura A.



Figura A: minimal gain:0.1, maximal depth 10 (invariata)

La Figura B invece è un esempio di un aumento della minimal gain. Come si può vedere L’albero è più piccolo rispetto al valore di default (0.01).

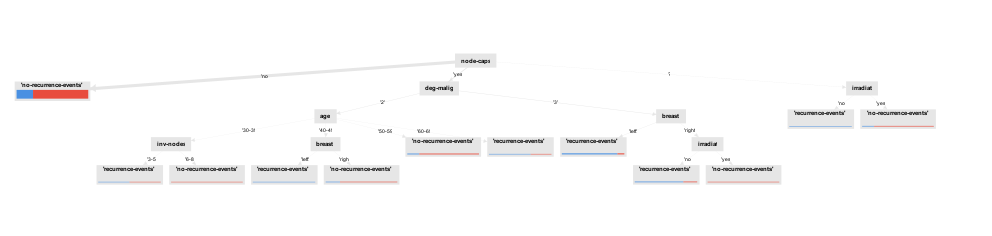


Figura B: minimal gain:0.04, maximal depth 10 (invariata)

Invece il parametro maximal depth permette di specificare l’altezza massima dell’albero. Se settiamo un valore inferiore all’altezza dell’albero quest’ultimo verrà tagliato all’altezza che abbiamo inserito. La Figura C ne è un esempio.

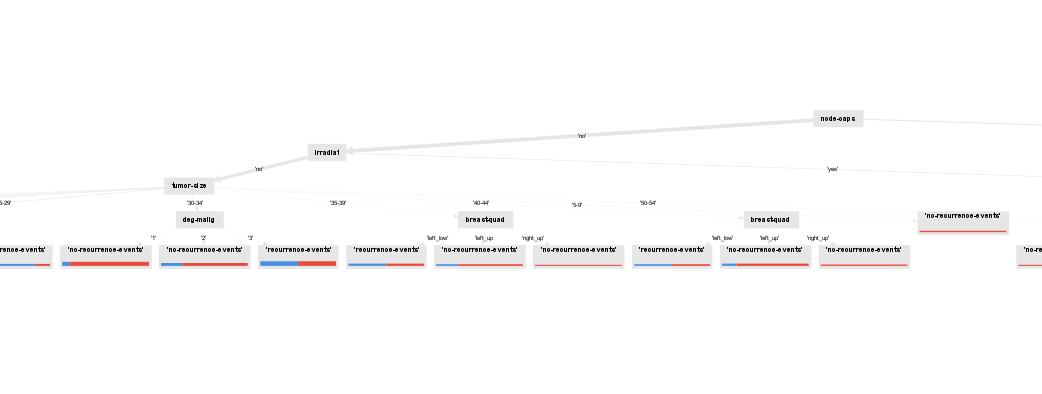


Figura C: minimal gain:0.01 (invariato), maximal depth: 5.

Avendo settato la maximal depth a 5 l’albero non può scendere ulteriormente. Invece, se settiamo maximal depth ad un valore superiore o uguale all’altezza dell’albero questo non subirà alcuna modifica, come si vede nella Figura D.

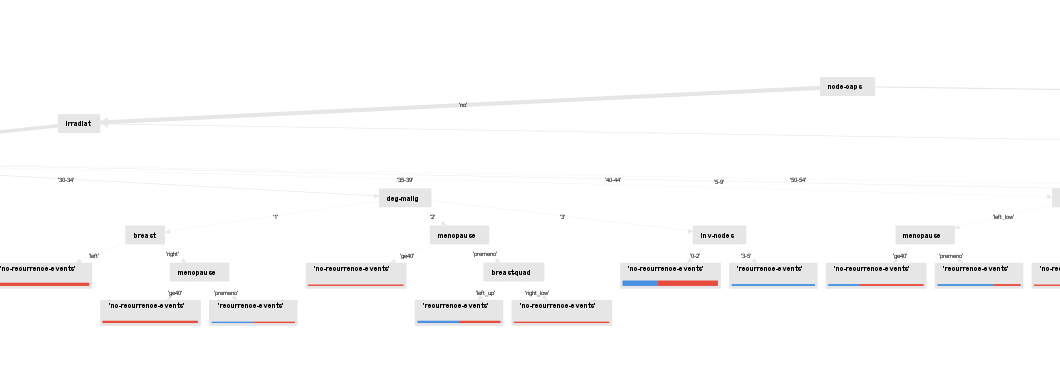


Figura D: minimal gain:0.01 (invariato), maximal depth:7.

Combinando entrambi i valori di maximal depth e minimal gain possiamo ottenere il risultato che più ci è utile.

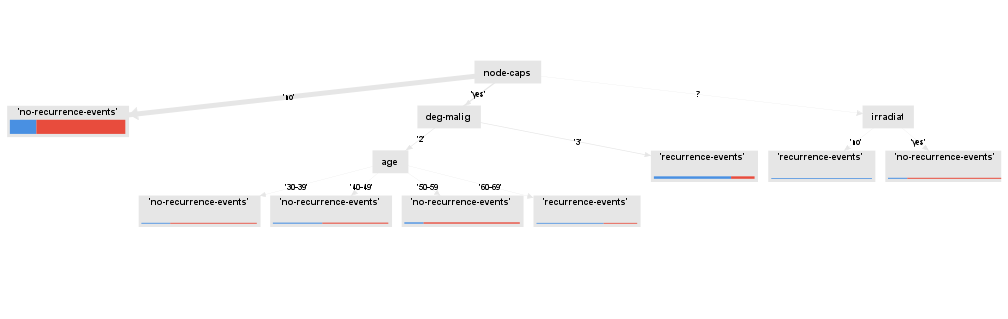
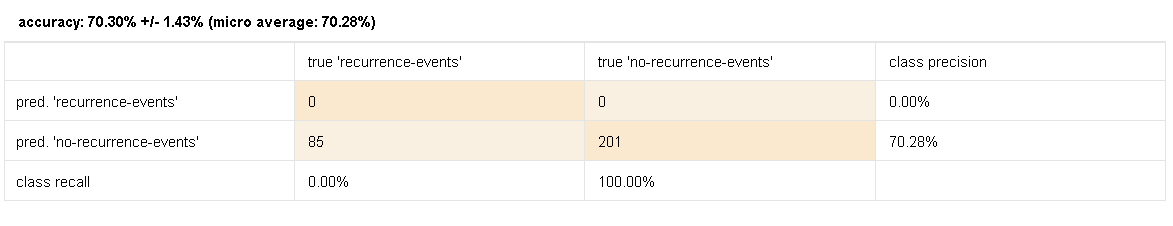


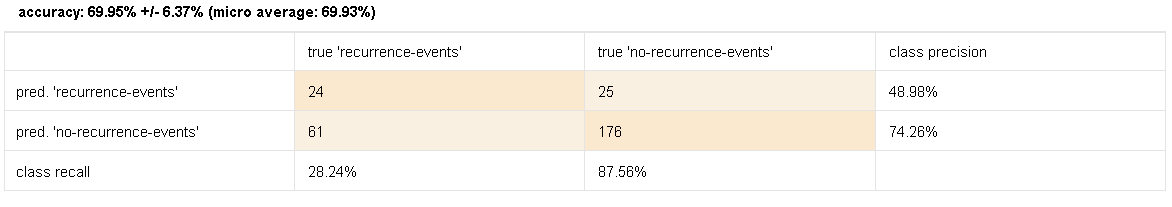
Figura E: minimal gain:0.04, maximal depth:4

**Domanda 3**

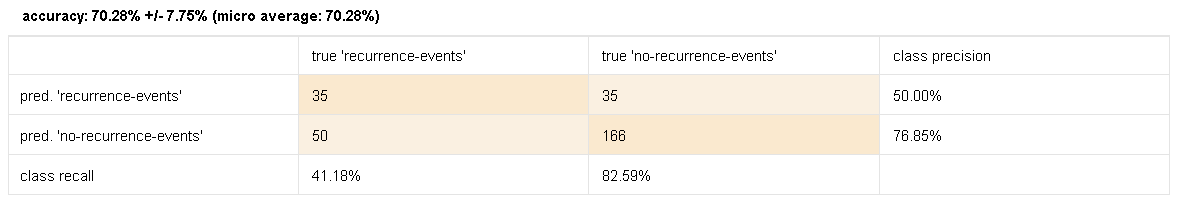
Riducendo il valore di minimal gain e aumentando la maximal depth l’accuratezza dovrebbe aumentare, ma come notiamo dagli screenshot con valori troppo bassi si può causare il problema dell’overfitting, che porta il modello a non funzionare bene con nuovi dati di test.



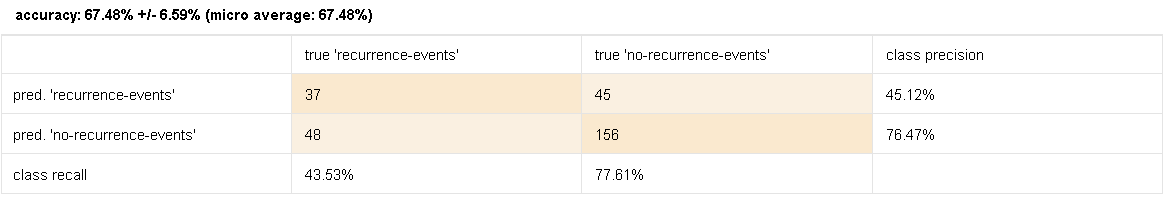
Minimal gain:0.1, maximal depth 10



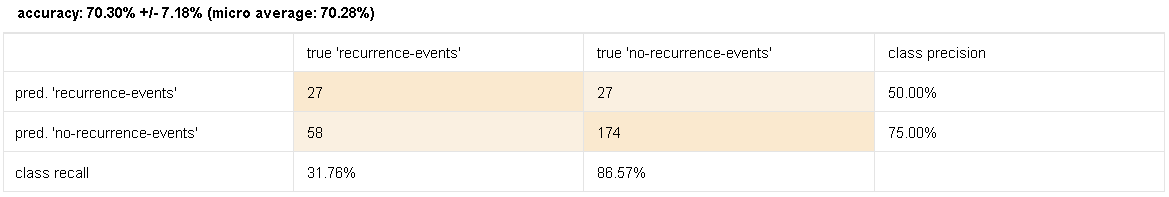
Minimal gain:0.04, maximal depth:10



Minimal gain:0.01, maximal depth:5



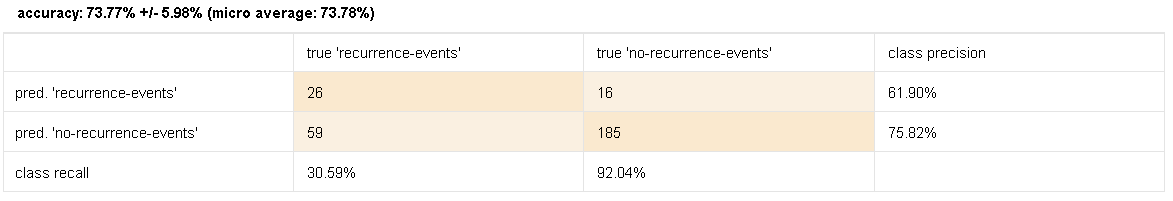
Minimal gain:0.01, maximal depth:7



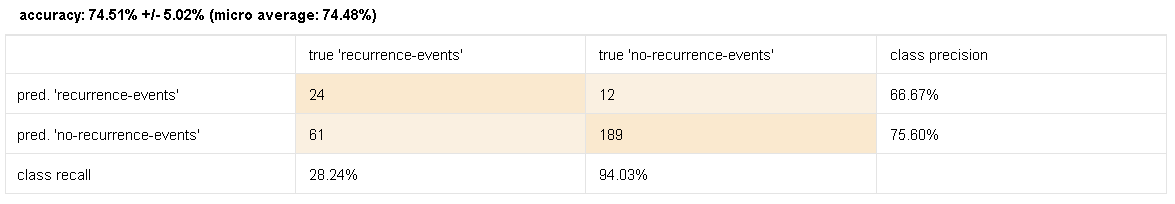
Minimal gain:0.04, maximal depth: 4

**Domanda 4**

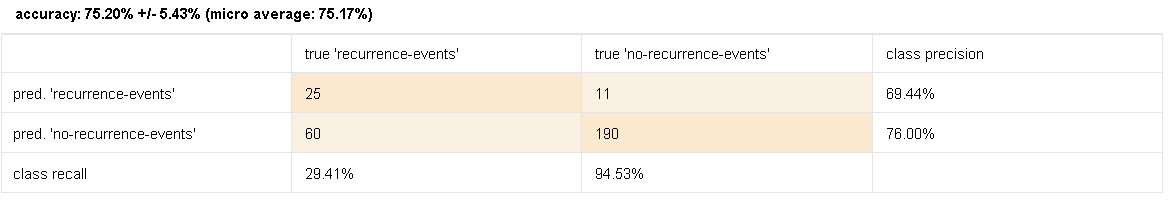
k-NN: Incrementando il valore di k, il classificatore andrà a considerare più dati vicini al quello di test, facendo così aumentare l’accuratezza del risultato. Superato però il valore 10 di k, i dati rumorosi iniziano ad inficiare sulla performance facendo diminuire l’accuratezza della classificazione. L’accuratezza di Naive Bayes risulta essere inferiore al K-NN con k=5, quindi possiamo concludere che convenga utilizzare l’algoritmo K-NN al posto di Bayes.

****

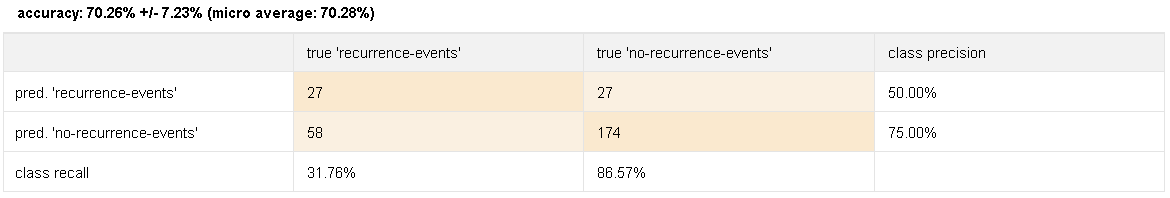
K=5



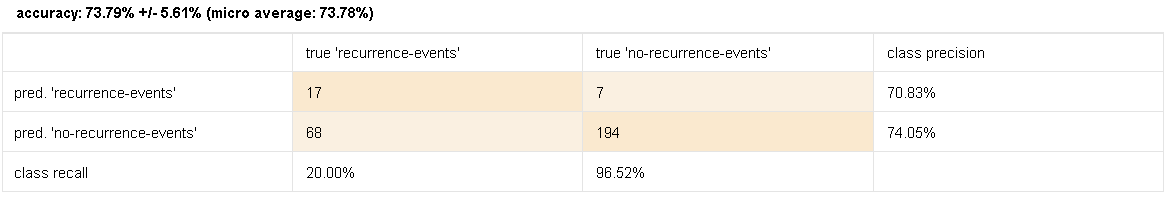
K=8



K=10

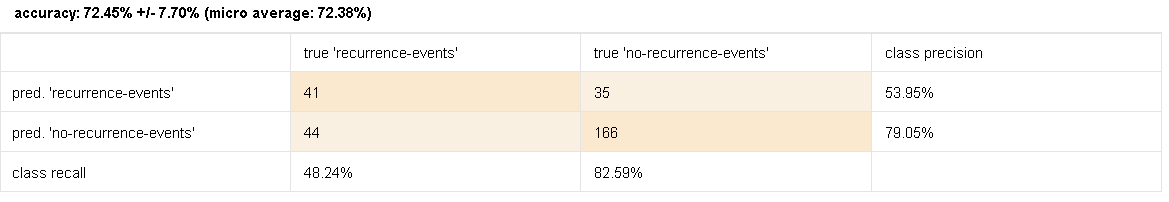


K=13



K=20

BAYES



**Domanda 5**

Matrice di correlazione. L’ipotesi Naive, ovvero l’indipendenza statistica, risulta abbastanza irrealistica, in quanto possiamo notare dei valori elevati per alcune coppie di attributi. Proprio per questa l’accuratezza di Naive Bayes risulta essere comparativamente bassa. I due attributi più correlati risultano essere “irradiat” e “inv-nodes”.

