Progetto Data Mining

Tommaso Mandoloni
tommaso.mandoloni@studio.unibo.it

Luglio 2022

Indice

Abstract

Scopo del progetto è analizzare un dataset attraverso tecniche di data mining, da cui ricavare un modello che permetta di risolvere task di classificazione binaria. Il flusso di lavoro è organizzato seguendo la metodologia CRISP-DM che permette di suddividere in maniera chiara e precisa le diverse fasi di progetto. Il dataset preso in considerazione è il Wisconsin Breast Cancer Database (`https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+%28original%29`), che quindi tratta dei tumori al seno in ambito sanitario, ed è composto da record di pazienti di cui viene descritto il profilo sanitario attraverso attributi specifici, e dei quali si vuole catalogare lo status di salute attuale. Verrà inoltre utilizzato WEKA come strumento di supporto durante il progetto di analisi.

1 Business Understanding

Come prima fase è importante definire il problema di data mining da risolvere, concentrandosi sui requisiti e obiettivi di progetto che si vuole raggiungere, anche sotto una visione di business. In particolare, analizzando il profilo sanitario di pazienti che presentano un tumore al seno, si vuole realizzare e addestrare un modello di data mining che permetta di agevolare il processo decisionale all'interno di un ospedale. Infatti, a fronte di alcune caratteristiche, o sintomi più tecnicamente parlando, che presentano le pazienti sottoposte ad una visita, il modello deve poter etichettare il tumore al seno di una paziente come benigno o maligno. In questo modo, nel caso in cui ci sia la possibilità della presenza di un tumore, il modello, se correttamente addestrato, potrebbe fornire una prima diagnosi della paziente, permettendo ai medici di essere al corrente della gravità del tumore anche prima di svolgere tutti gli accertamenti del caso, che porterebbero ad un ulteriore costo di tempo e risorse (meglio prevenire che curare).

2 Data Understanding

Il task da svolgere è quindi un'attività di classificazione binaria (tumore benigno o tumore maligno) attraverso l'addestramento di un modello sulla base di record forniti come input, e che quindi sappia generalizzare anche su dati sconosciuti, in modo da poter essere utilizzato come strumento di supporto alle diagnosi mediche.

Il dataset delle pazienti è formato da 699 record con i seguenti attributi che li descrivono (seguiti fra parentesi dal dominio dei valori che possono acquisire):

- CodeNumber: codice identificativo della paziente sottoposta alla visita (ID incrementale).
- ClumpThickness: livello di spessore del grumo, che indica il raggruppamento di cellule tumorali nel multistrato (1 - 10).

- UniformityCellSize: livello di uniformità della dimensione cellulare, indica la metastasi ai linfonodi, ossia lo spostamento delle cellule tumorali (1 - 10).
- UniformityCellShape: livello di uniformità delle forme cellulari, identificando cellule cancerose di dimensioni variabili (1 - 10).
- MarginalAdhesion: livello di adesione marginale, ovvero una perdita di adesione, cioè un segno di malignità (1 - 10).
- SingleEpithelialCellSize: livello di dimensione della singola cellula epiteliale, quindi, se diventa grande, potrebbe essere una cellula maligna (1 - 10).
- BareNuclei: livello di nuclei nudi, senza rivestimento citoplasmatico, che si trovano nei tumori benigni (1 - 10).
- BlandChromatin: livello di cromatina blanda, solitamente presente nelle cellule benigne (1 - 10).
- NormalNucleoli: livello di nucleoli normali, generalmente molto piccoli nelle cellule benigne (1 - 10).
- Mitoses: livello della mitosi, il processo di divisione cellulare mediante il quale il nucleo si divide (1 - 10).
- Class: attributo target che indica se il tumore è benigno o maligno (2 - benigno, 4 - maligno).

Features	ClumpThickness	Numeric	1-10
	UniformityCellSize	Numeric	1-10
	UniformityCellShape	Numeric	1-10
	MarginalAdhesion	Numeric	1-10
	SingleEpithelialCellSize	Numeric	1-10
	BareNuclei	Numeric	1-10
	BlandChromatin	Numeric	1-10
	NormalNucleoli	Numeric	1-10
	Mitoses	Numeric	1-10
	Class	Nominal	Benign-Malignant
Class Distribution		Benign: 458 (65.5%)	
		Malignant: 241 (34.5%)	
Missing Values		16	
Instances		699	

Figure 1: Schema riassuntivo del dataset.

Per prima cosa è importante visualizzare l'intero dataset in relazione all'attributo classe (target) per poter avere subito una visione globale dei dati. Aprendo il dataset in WEKA ci si accorge subito che l'attributo Class è numerico, e quindi

non si riesce ad avere una chiara visione di come viene distribuito rispetto agli altri attributi. Perciò è necessaria una conversione da attributo numerico a nominale, in cui invece di avere le label delle classi numeriche, vengono anche rinominate per aumentare la chiarezza. Per ottenere una visualizzazione più chiara della classe quindi si è scelto di utilizzare un filtro `Discretize` direttamente sull'attributo 'Class' in modo da trasformarlo in nominale ed essere sicuri che venga appunto discretizzato in due valori trattandosi di un problema di classificazione binaria. Una volta applicato il filtro e rinominati le label della classe, il risultato è il seguente:

A questo è possibile avere una visione generale dell'intero dataset, potendo visualizzare come l'attributo classe viene distribuito rispetto agli altri attributi, per poter subito fare delle prime considerazioni riguardo il task che si vuole risolvere.

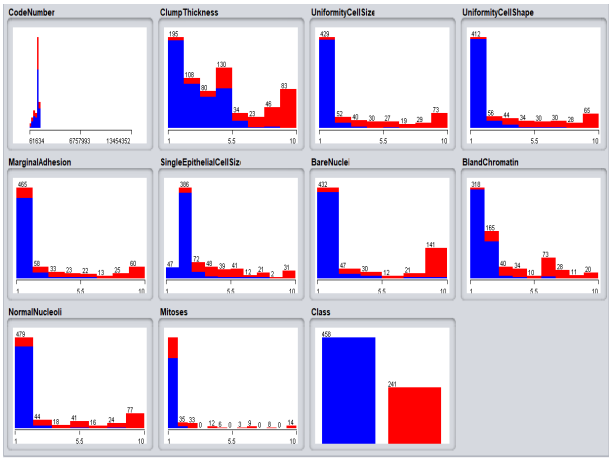


Figure 2: Panoramica della distribuzione monodimensionale dell'attributo 'Class' rispetto agli attributi.

Ciò che emerge da questa prima analisi è che valori bassi degli attributi implicano l'appartenenza del record alla classe 'benign', e infatti si può notare come il colore blu sia più presente rispetto al rosso nelle prime colonne degli istogrammi. Al contrario il rosso emerge con più decisione all'aumentare dei valori degli attributi, il che indica l'appartenenza del record alla classe 'malignant'.

Il prossimo passo è capire che relazione c'è fra gli attributi del dataset, ossia capire se esiste un qualche tipo di correlazione che permetta di ricavare informazioni importanti sui dati. In particolare è utile visualizzare dei piani cartesiani in cui vengono fissati gli attributi sugli assi per visualizzare come i dati vengono distribuiti al variare dei valori di uno o dell'altro. Da un'analisi generale, aggiungendo del rumore per visualizzare meglio i dati, è emerso che tutti i grafici di correlazione tra gli attributi presentano all'incirca lo stesso pattern riguardo la distribuzione dei dati rispetto alla classe e agli attributi stessi.

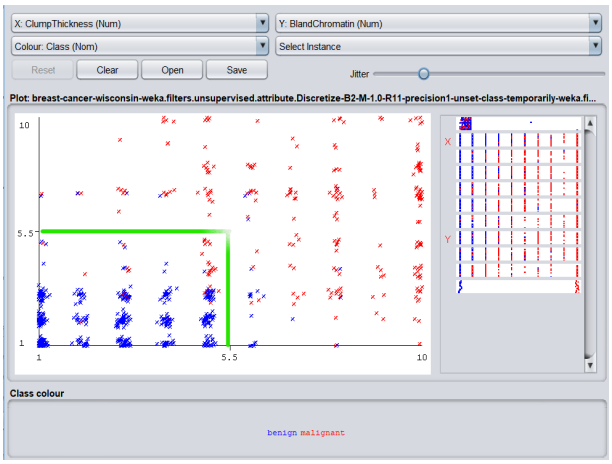


Figure 3: Visualizzazione del pattern di distribuzione dei dati.

La linea verde nella figura indica la zona in cui i dati risultano essere della classe 'benign' mentre al di fuori di questa zona i dati sono classificati per lo più come 'malignant'. Questa informazione rispetta esattamente l'analisi preliminare effettuata in precedenza in quanto anche in questo caso emerge che per valori bassi di entrambi gli attributi la classe del record sarà 'benign' mentre con l'aumentare dei valori di uno o dell'altro, il record sarà di classe 'malignant'.

In particolare, analizzando le varie combinazioni di attributi, sono emerse delle caratteristiche importanti:

- L'attributo 'BareNuclei' è molto importante per l'analisi, in quanto per valori alti dell'attributo corrispondono sempre numerosi casi di tumore maligno, a prescindere da quale sia l'attributo con il quale è messo in relazione. Questo attributo quindi porta con se un informazione molto importante e che risulta determinante ai fini dell'analisi, poiché una paziente che presenta medio/alti valori di questo attributo è molto probabile che abbia un tumore maligno.

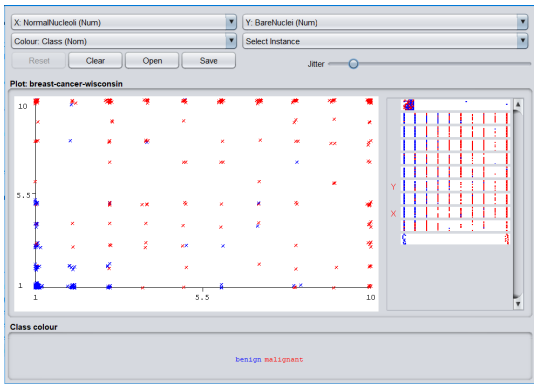


Figure 4: E' possibile osservare come per valori alti del livello di nuclei nudi, le istanze di classe 'malignant' siano molto numerose, anche se i valori del livello di nucleoli normali è medio/basso.

- L'attributo 'Mitoses' sembra non essere troppo determinante per l'analisi in quanto presenta solo valori medio/bassi. Per questo motivo non saranno molti i record che possiedono valori alti di questo attributo, e, quando messo in relazione con un altro, non avrà molto impatto sulla classificazione dell'istanza, dato che la classe dell'istanza sarà determinata soltanto dall'attributo con cui è messo in relazione (nei casi più estremi il modello potrebbe considerare i valori alti come outlier).

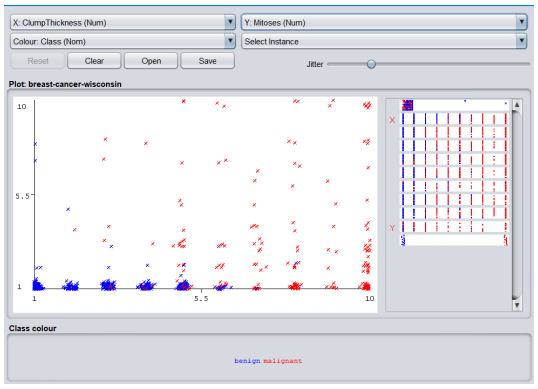


Figure 5: E' possibile notare come l'attributo in questione non superi un certa soglia di valori, e quindi la classe viene determinata per lo più dall'attributo con cui si relaziona.

3 Data Preparation

In questa sezione vengono descritte le operazioni a livello di dati che sono state eseguite per preparare il dataset, in modo da facilitare il modello nell'addestramento.

3.1 Feature Selection

E' importante selezionare soltanto gli attributi che portano informazione determinante perciò è stato immediatamente rimosso l'ID delle pazienti, poiché appunto non permetterebbe al modello di generalizzare su dati che non conosce, cioè non deve imparare informazioni sui dati sulla base di un attributo specifico come l'ID. Anche l'attributo 'Mitoses' discusso in precedenza potrebbe non risultare rilevante ai fini dell'analisi ma comunque potrebbe portare informazione per quanto riguarda i valori medio/bassi. L'ipotesi è quindi contrastante poiché da una parte sembra che possa essere rimosso rimosso in quanto poco rilevante mentre dall'altra potrebbe comunque portare un guadagno informativo se pur minimo.

Per valutare la correttezza delle scelte proposte è stato utilizzato l'algoritmo di *attribute selection* presente in WEKA `CfsSubsetEval` con metodo di ricerca `BestFirst`, che ha prodotto i risultati attesi, ossia ha rimosso l'attributo dell'ID e mantenuto 'Mitoses' nonostante abbia pochi valori alti. Per ottenere risultati migliori su quali attributi selezionare per addestrare il modello, è stata effettuata ance una prova con l'algoritmo `InfoGainAttributeEval` con metodo di ricerca `Ranker` che permette di valutare il guadagno informativo di ogni attributo rispetto alla classe e di stilare un elenco decrescente degli attributi più rilevanti.

```
Ranked attributes:
 0.675    2 UniformityCellSize
 0.66     3 UniformityCellShape
 0.564    6 BareNuclei
 0.543    7 BlandChromatin
 0.505    5 SingleEpithelialCellSize
 0.466    8 NormalNucleoli
 0.459    1 ClumpThickness
 0.443    4 MarginalAdhesion
 0.198    9 Mitoses
```

Figure 6: Ranking del guadagno informativo degli attributi rispetto alla classe.

Dal risultato emerge che l'attributo 'Mitoses' è in ultima posizione con un valore di *gain* piuttosto ridotto rispetto al resto degli attributi. Anche in questo caso l'ipotesi iniziale sembra rispettata in quanto l'attributo porta informazione ma in maniera quasi irrilevante rispetto agli altri attributi.

In ogni caso verranno analizzate le performance dei modelli sia considerando

l'attributo 'Mitoses' all'interno del dataset, sia rimuovendolo, in modo da valutare entrambe le opzioni.

3.2 Outliers Detection

La presenza di outlier in un ambito così delicato potrebbe portare il modello a non essere del tutto accurato, perciò è stata effettuata un'analisi dei valori degli attributi per individuarne la presenza. Da una prima osservazione del dataset, gli attributi hanno tutti valori compresi tra 1 e 10 interi, perciò non ci si aspetta che ci siano outlier che escano dal range di questi valori, ma è stato comunque utilizzato l'algoritmo di *outlier detection* `InterQuantileRange` proposto da WEKA per confermare o meno questa ipotesi. Sostanzialmente l'algoritmo si basa su calcoli statistici, ovvero calcola l'IQR (inter quantile range) di ogni attributo che, messo in relazione con il primo e terzo quantile (25esimo e 75esimo percentile rispettivamente) della distribuzione dei dati per ogni valore, determina se quel dato potrebbe o meno essere un outlier.

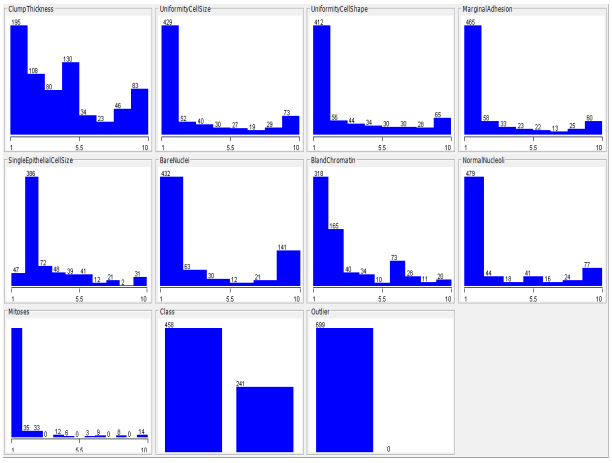


Figure 7: Distribuzione di valori considerati come *outlier* tra gli attributi.

Come ci si aspettava dall'analisi preliminare, non sono presenti outlier all'interno del dataset e quindi anche il modello potrebbe trarne vantaggio a livello di accuratezza.

3.3 Missing Values

L'attributo 'BareNucleoli', se pur in quantità minime (16), dei valori mancanti che potrebbero quindi non portare l'informazione necessaria al modello per poter classificare il dato correttamente. Perciò è stata effettuata una operazione di *replacement* attraverso WEKA (`ReplaceMissingValules`) in cui i valori mancanti venivano sostituiti con la media dei valori (3.545), e alla quale è stata poi effettuata un'operazione di *flooring* per riportare il valore intero, e quindi

coerente con il resto dei valori presenti nel dataset. Le operazioni di rimpiazzo e arrotondamento hanno quindi sostituito tutti i valori mancanti presenti con il valore 4. E' stata effettuata un'operazione di arrotondamento piuttosto che di approssimazione all'intero precedente o successivo perché comunque sia il 3 che il 4 rientrano tra i valori medio/bassi dell'attributo e che quindi portano all'incirca la stessa informazione. Sarebbe stato diverso per valori confinanti come il 5 o il 6 che se arrotondati per eccesso potevo portare un'informazione diversa che se arrotondati per difetto. In realtà non è detta che questa operazione sia ottimale, in quanto la dimensione di una singola cella non è per forza correlata con la dimensione media delle altre celle. Trattandosi però di un numero di record molto bassi, adottare un'operazione di *replacing* o una di *removing* (attraverso `RemoveWithValues` di WEKA) non porta cambiamenti sostanziali alle performance del modello, ma verranno comunque prese in considerazione entrambe le soluzioni per poter comparare i risultati.

3.4 Feature Transformation

La fase di *feature transformation* riguarda soltanto la distribuzione dei valori degli attributi, in quanto poiché i valori originali avevano già delle caratteristiche ben definite e coerenti, ossia erano tutti interi compresi tra 1 e 10, non c'è stato bisogno di effettuare operazioni particolari. L'unico accorgimento, proprio perché si conosceva il range di valori degli attributi, è stato quello di discretizzarli in 10 bins tramite il filtro `Discretize`, in modo che ogni bin contenesse soltanto i record di un determinato valore, cercando di conferire al dataset maggiore espressività.

4 Modeling

Come si è potuto osservare durante l'analisi del dataset, le due classi sono sbilanciate, perciò è necessario utilizzare modelli di machine learning *insensitive* a questo tipo di situazione.

4.1 Performance

Per valutare le prestazioni dei modelli utilizzati sono stati presi in considerazione i seguenti indici:

- **Accuracy**: è l'indicatore (KPI) principale che verrà utilizzato in fase di valutazione del modello, in quanto rappresenta il rapporto di tutti i record classificati correttamente rispetto a tutti i record presenti nel dataset.
- **Precision**: esprime il rapporto tra il numero delle previsioni corrette di un evento (classe) sul totale delle volte che il modello lo prevede, tra cui possono esserci degli errori. E' importante quindi controllare che il valore di `precision` non sia basso, poiché significherebbe che il classificatore associa la classe 'benign' a un record che in realtà presenta un tumore

maligno, ovvero l'errore più grave che il classificatore possa compiere (FP) [1].

- **Recall**: esprime il rapporto tra le previsioni corrette per una classe sul totale dei casi in cui si verifica effettivamente e che può anche non aver riconosciuto. E' importante in questo caso che la `recall` sia molto elevata, ciò significherebbe che il classificatore non commette molti errori quando deve associare un dato ad un classe, e perciò il numero di istanze classificate come 'malignant', ma che in realtà sono di classe 'benign', sarà molto basso (FN) [1].

- **ROC Curve (AUC)**: è un grafico che mette in relazione la **sensibilità** e la **specificità** di un test diagnostico. La sensibilità indica il rapporto tra i record di classe 'benign' classificati correttamente e tutti quelli realmente 'benign', mentre la specificità indica il rapporto tra i record classificati correttamente come 'malignant' e tutti quelli realmente 'malignant'. In generale è da preferire un test sensibile quando l'individuazione di una malattia che in realtà non è presente ha conseguenze dannose (vengono effettuate visite di controllo o interventi non necessari), mentre è da preferire un test specifico quando la mancata individuazione di una malattia può avere conseguenze pericolose (non viene presa in considerazione l'eventuale presenza di un tumore maligno).

- **MCC** (Matthews Correlation Coefficient): coefficiente di correlazione tra classi e previsioni. In genere varia tra -1, quando c'è un perfetto disaccordo tra valori effettivi e previsioni (il modello ha previsto erroneamente la classe opposta ogni volta), e +1, quando c'è un perfetto accordo tra valori effettivi e previsioni (il modello ha previsto correttamente la classe ogni volta). 0 quando la previsione può anche essere casuale rispetto ai valori reali (il modello genera una congettura casuale). Poiché coinvolge i valori di tutti e quattro i quadranti di una matrice di confusione, è considerata una misura bilanciata. [2]

R \ P	benign	malignant
benign	TP	FN
malignant	FP	TN

Figure 8: Matrice di confusione di riferimento.

Ogni esperimento effettuato sul modello, viene testato prima sulle istanze del training set, e comparato con una *k-fold cross validation* con $k = 10$, in modo

[1] In particolare, per questo tipo di dominio applicativo, è estremamente importante prestare attenzione soprattutto ai falsi negativi (FN) e ai falsi positivi (FP), in quanto non si dovrebbe iniziare il processo delle visite di controllo per un presunto tumore maligno quando in realtà non lo presenta realmente (risparmio di tempo e costi), e viceversa è fondamentale non ignorare un paziente che sembra non avere un tumore maligno ma in realtà è in pericolo e bisogna subito intervenire con i controlli necessari.

[2] Un caso d'uso esemplificativo è disponibile al seguente link `https://lettier.github.io/posts/2016-08-05-matthews-correlation-coefficient.html`

da testare il modello in maniera più precisa sfruttando il *validation set*, facendo quindi previsioni su dati che non conosce ad ogni iterazione. Così facendo si ha più controllo sull'**overfitting** del modello, cercando quindi di evitare che impari a classificare correttamente soltanto le istanze del *training set*, e riesca a generalizzare bene anche su dati che non conosce.

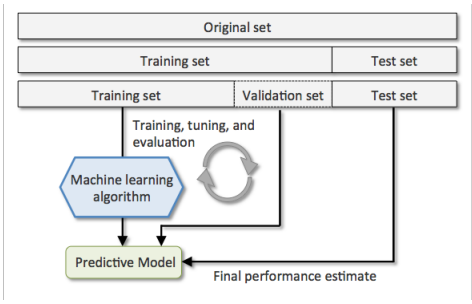
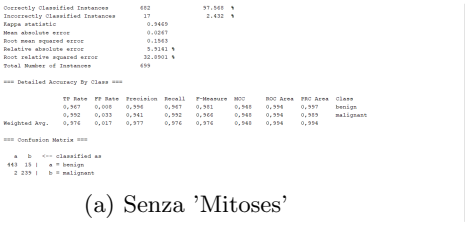


Figure 9: Addestramento modello utilizzando *cross validation*.

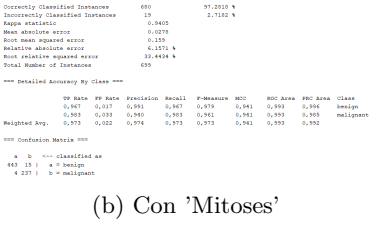
Naive Bayes Classifier

Lavora molto bene su dataset con classi sbilanciate, ma occorre che non siano presenti valori mancanti e gli attributi siano di tipo nominale. Si tratta quindi di un classificatore probabilistico che si basa sulla regola $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$, ovvero la probabilità di ottenere A dato B. In relazione al problema che si sta analizzando, si vuole trovare la probabilità che un dato appartenga a una certa classe sulla base dei valori dei suoi attributi, e quindi la regola diventa $P(y|x_1, x_2, .., x_9) = \frac{P(x_1|y)P(x_2|y)..P(x_9|y)P(y)}{P(x_1)P(x_2)..P(x_9)}$, dove $x_1, x_2, .., x_9$ sono i valori dei 9 attributi, mentre y è la classe. E' stato quindi addestrato un modello bayesiano suddividendo sia la casistica in cui venga considerato l'attributo 'Mitoses' o meno, sia la modalità di gestione dei missing values ('replace' o 'remove').

Nel caso in cui venga adottata una politica di *replacing* per i missing values, l'addestramento del modello produce pressoché lo stesso risultato con qualche minima sfaccettatura dovuta al fatto che venga o meno considerato l'attributo 'Mitoses'.



(a) Senza 'Mitoses'



(b) Con 'Mitoses'

Figure 10: Confronto risultati addestramento [replace].

Inoltre è stato eseguito l'addestramento sia utilizzando il training set per

il testing, sia sfruttando la *cross-validation* (10 fold) e anche in questo caso produceva gli stessi risultati. Di conseguenza si potrebbe dedurre che il modello, oltre che ad essere molto preciso, generalizza bene e non cade in una condizione di *overfitting*.

Nel caso in cui venga scelta una politica di *removing* per il trattamento dei missing values, l'addestramento produce risultati molto simili sia al caso precedente, sia al fatto che venga o meno utilizzato l'attributo 'Mitoses'.

(a) Senza 'Mitoses' (b) Con 'Mitoses'

Figure 11: Confronto risultati addestramento [remove].

Anche in questo caso, effettuando l'addestramento e testando i risultati sul training i set, i risultati erano gli stessi perciò il modello riesce a generalizzare molto bene.

In generale quindi, un modello di questo tipo riesce a gestire la classificazione in maniera quasi ottimale. I risultati prodotti presentano valori vicini all'ottimo per quanto riguarda gli indici di valutazione scelti, indicando robustezza e precisione del modello, e quindi anche un ottimo bilanciamento tra TPR e FPR.

	Discretize + Replace Missing Values			Discretize + Remove Missing Values		
Attributes	Accuracy	FN	FP	Accuracy	FN	FP
8 + Class	97.56%	15	2	97.65%	13	3
9 + Class	97.28%	15	4	97.36%	13	5

Figure 12: Tabella riassuntiva risultati Naive Bayes.

Inoltre, visualizzando la tabella, è importante notare come il modello compie più errori per quanto riguarda i False Negative rispetto ai False Positive. Questo significa che il modello è più probabile che associ la classe 'malignant' ad una paziente che in realtà ha un tumore benigno. Ovviamente si tratta sempre di un errore e quindi implica costi aggiuntivi per le visite, ma comunque è un errore meno grave rispetto alla situazione inversa in cui associa la classe 'benign' ad una paziente che in realtà ha un tumore maligno, rischiando quindi di ignorare il problema.

J48 Tree Classifier

Molto performante per dataset con classi sbilanciate e riesce ad operare anche in caso di valori mancanti. Con il crescere dell'albero può accadere che il modello vada inevitabilmente a finire in una situazione di *overfitting*, perciò sarà

necessario valutare il problema effettuando o meno un'operazione di **pruning** dell'albero. Per prima cosa è stato utilizzato il modello con i parametri di default (effettuando **pruning** dell'albero), che ha prodotto ottimi risultati in termini di accuratezza, ma rischiava di essere troppo erroneo per quanto riguarda i FP, che bisogna ricordare di tenere sotto controllo. Inoltre, visualizzando l'albero prodotto dal modello, risulta essere molto confusionario e complesso. Perciò si è scelto di andare ad effettuare un'operazione di *fine-tuning* dei parametri dell'algoritmo in modo da trovare la combinazione che producesse i risultati migliori. Dopo un certo numero di prove, si è visto che il modello, oltre ad aumentare la sua accuratezza se pur lievemente, effettuava meno errori per quanto riguarda i FP. La combinazione di parametri che si è scelta quindi è quella che descrive un albero decisionale che faccia `binary-split` degli attributi (e non che consideri tutti i valori possibili) effettuando un'operazione di pruning con *fattore di confidenza* 0.1 [3].

```
  a   b   <-- classified as
436  22 |   a = benign
 23 218 |   b = malignant
```

(a) J48 Default

```
  a   b   <-- classified as
431  27 |   a = benign
 12 229 |   b = malignant
```

(b) J48 Parametrizzato

Figure 13: Confronto delle confusion-matrix.

Anche l'albero decisionale prodotto dall'algoritmo sembra essersi semplificato, garantendo inoltre maggiore accuratezza nei risultati.

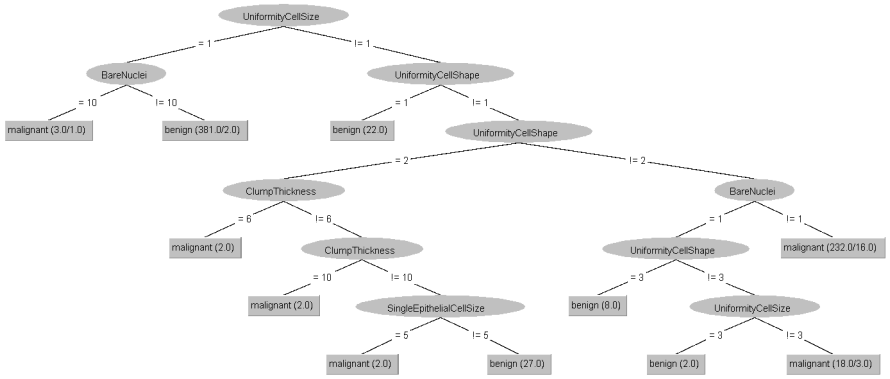
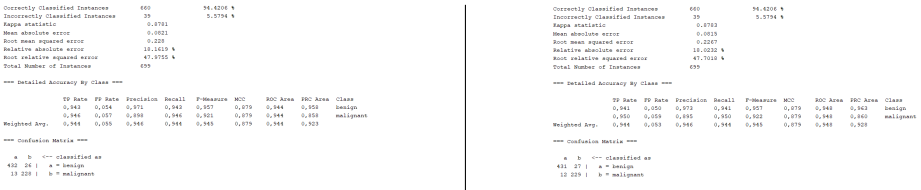


Figure 14: Albero Decisionale

Non effettuando l'operazione di *pruning* dell'albero, le prestazioni peggioravano (aumentano gli errori per quanto riguarda FP e l'accuratezza cala), perciò si è deciso di adottare il modello con i parametri ottimali descritti sopra per

[3] fattore di pruning dell'albero: più il numero è basso, più l'albero tende a non crescere

effettuare gli esperimenti di confronto successivi. Quindi, nel caso sia stata adottata una politica di *replacing* di valori mancanti, il modello ha prodotto i seguenti risultati.

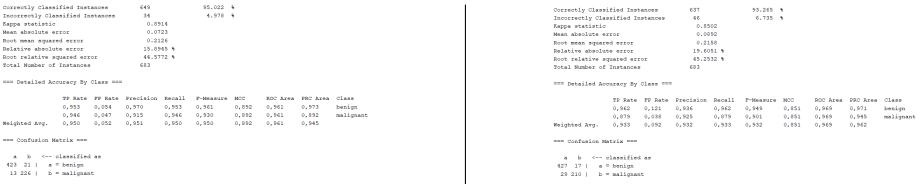


(a) Senza 'Mitoses'

(b) Con 'Mitoses'

Figure 15: Confronto risultati addestramento [replace].

Mentre, nel caso sia stata effettuata una rimozione dei missing values, sono stati prodotti i seguenti risultati.



(a) Senza 'Mitoses'

(b) Con 'Mitoses'

Figure 16: Confronto risultati addestramento [remove].

Anche in questo caso, si riescono a generalizzare molto bene le informazioni che il modello apprende dai dati, infatti i livelli di accuratezza scaturiti dagli esperimenti sono ottimi. In particolare, nel caso venga effettuata un'operazione di *replacing* dei valori mancanti, i risultati sono molto più simili tra loro in caso si consideri o meno l'attributo 'Mitoses', mentre nel caso vengano rimossi, si nota una leggera differenza.

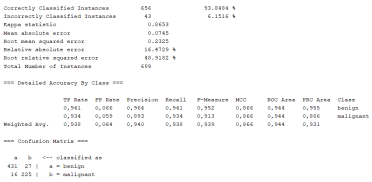
	Discretize + Replace Missing Values			Discretize + Remove Missing Values		
Attributes	Accuracy	FN	FP	Accuracy	FN	FP
8 + Class	94.42%	26	13	95.02%	21	13
9 + Class	94.42%	27	12	93.26%	17	29

Figure 17: Tabella riassuntiva risultati J48.

JRip Classifier

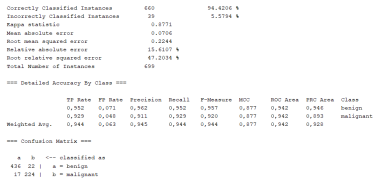
Per rafforzare (o confutare) i risultati ottenuti dall'albero decisionale, è stato utilizzato un modello di classificazione basato su **regole** che sfrutta l'algoritmo RIPPER.

Come per la costruzione dell'albero decisionale, anche in questo caso i risultati migliori sono stati raggiunti eseguendo un *pruning* delle regole, in modo da non generare regole insignificanti. L'unico parametro che è stato modificato tramite una fase di *fine-tuning* è stato il numero minimo di elementi coperti da una regola che è stato fissato a 1. Eseguendo gli stessi esperimenti dei modelli precedenti sono emersi i seguenti risultati.



```
Correctly Classified Instances         656               93.8484 %
Incorrectly Classified Instances        43                6.1516 %
Kappa statistic                          0.8653
Mean absolute error                      0.0745
Root mean squared error                  0.2325
Relative absolute error                 16.4729 %
Root relative squared error             49.5102 %
Total Number of Instances              699

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0,941     0,066     0,964     0,941     0,952      0,866     0,964     0,955     benign
                 0,934     0,059     0,893     0,934     0,913      0,866     0,964     0,896     malignant
Weighted Avg.    0,938     0,064     0,940     0,938     0,939      0,866     0,964     0,931

=== Confusion Matrix ===

   a   b   <-- classified as
 431  27 |   a = benign
  16 225 |   b = malignant
```

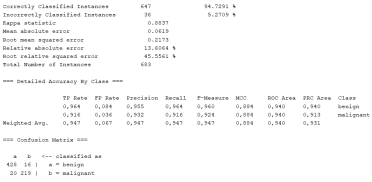
(a) Senza 'Mitoses'



```
Correctly Classified Instances         660               94.4206 %
Incorrectly Classified Instances        39                5.5794 %
Kappa statistic                          0.8771
Mean absolute error                      0.0706
Root mean squared error                  0.2244
Relative absolute error                 15.6107 %
Root relative squared error             47.2036 %
Total Number of Instances              699

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0,952     0,071     0,962     0,952     0,957      0,877     0,942     0,946     benign
                 0,929     0,048     0,911     0,929     0,920      0,877     0,942     0,893     malignant
Weighted Avg.    0,944     0,063     0,945     0,944     0,944      0,877     0,942     0,928

=== Confusion Matrix ===

   a   b   <-- classified as
 436  22 |   a = benign
  17 224 |   b = malignant
```

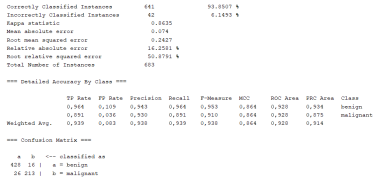
(b) Con 'Mitoses'

Figure 18: Confronto risultati addestramento [replace].



```
Correctly Classified Instances         647               94.7291 %
Incorrectly Classified Instances        36                5.2709 %
Kappa statistic                          0.8837
Mean absolute error                      0.0619
Root mean squared error                  0.2173
Relative absolute error                 13.6064 %
Root relative squared error             45.5561 %
Total Number of Instances              683

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0,964     0,084     0,955     0,964     0,960      0,884     0,940     0,940     benign
                 0,916     0,036     0,932     0,916     0,924      0,884     0,940     0,913     malignant
Weighted Avg.    0,947     0,067     0,947     0,947     0,947      0,884     0,940     0,931

=== Confusion Matrix ===

   a   b   <-- classified as
 428  16 |   a = benign
  20 219 |   b = malignant
```

(a) Senza 'Mitoses'



```
Correctly Classified Instances         641               93.8507 %
Incorrectly Classified Instances        42                6.1493 %
Kappa statistic                          0.8635
Mean absolute error                      0.074
Root mean squared error                  0.2427
Relative absolute error                 16.2581 %
Root relative squared error             50.8791 %
Total Number of Instances              683

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0,964     0,109     0,943     0,964     0,953      0,864     0,929     0,936     benign
                 0,891     0,036     0,930     0,891     0,910      0,864     0,929     0,875     malignant
Weighted Avg.    0,939     0,083     0,938     0,939     0,938      0,864     0,929     0,914

=== Confusion Matrix ===

   a   b   <-- classified as
 428  16 |   a = benign
  26 213 |   b = malignant
```

(b) Con 'Mitoses'

Figure 19: Confronto risultati addestramento [remove].

In questo caso, nonostante si ottenga comunque un livello di accuratezza piuttosto buono, gli errori per quanto riguarda FN e FP sono abbastanza numerosi ma molto simili a quelli prodotti dall'albero decisionale, come ci si poteva aspettare. D'altro canto però, anche gli altri indici di valutazione (MCC, AUC) esprimono valori vicini all'ottimo, facendo intuire che il modello riesce comunque a gestire in maniera adeguata il problema. Il classificatore ha prodotto le seguenti regole:

```
(BareNuclei = 10) => Class=malignant (132.0/3.0)
(NormalNucleoli = 10) => Class=malignant (32.0/0.0)
(UniformityCellSize = 10) => Class=malignant (18.0/0.0)
(BlandChromatin = 7) and (SingleEpithelialCellSize = 3) => Class=malignant (6.0/0.0)
(UniformityCellSize = 8) => Class=malignant (11.0/1.0)
(ClumpThickness = 10) => Class=malignant (12.0/0.0)
(UniformityCellShape = 4) and (SingleEpithelialCellSize = 3) => Class=malignant (3.0/0.0)
(MarginalAdhesion = 4) and (BlandChromatin = 4) => Class=malignant (4.0/0.0)
(UniformityCellSize = 6) => Class=malignant (5.0/0.0)
(UniformityCellSize = 3) and (BareNuclei = 4) => Class=malignant (2.0/0.0)
(MarginalAdhesion = 10) => Class=malignant (4.0/1.0)
(UniformityCellSize = 5) => Class=malignant (4.0/0.0)
(UniformityCellSize = 3) and (NormalNucleoli = 9) => Class=malignant (2.0/0.0)
(SingleEpithelialCellSize = 6) => Class=malignant (3.0/1.0)
(NormalNucleoli = 4) => Class=malignant (3.0/1.0)
 => Class=benign (442.0/5.0)

Number of Rules : 16
```

Figure 20: Regole Decisionali

Confrontando l'insieme delle regole con l'albero decisionale si può notare che vengono rispettate circa le stesse decisioni riguardo l'importanza degli attributi. Ad esempio, come nell'albero decisionale viene controllato già nel secondo livello se l'attributo `BareNuclei` sia uguale a 10, anche nelle regole prodotte dal classificatore `BareNculei` uguale a 10 è la prima condizione che viene controllata.

	Discretize + Replace Missing Values			Discretize + Remove Missing Values		
Attributes	Accuracy	FN	FP	Accuracy	FN	FP
8 + Class	93.84%	27	16	94.72%	16	20
9 + Class	94.42%	22	17	93.85%	16	26

Figure 21: Tabella riassuntiva risultati JRip.

5 Evaluation

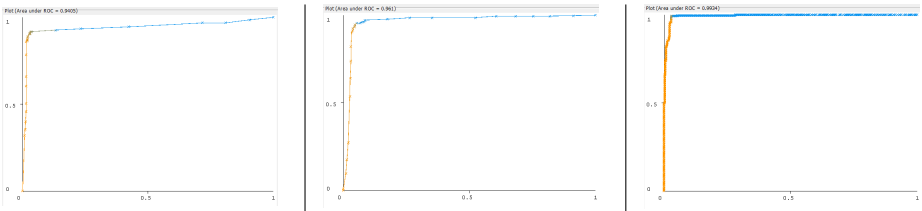
Per effettuare una corretta valutazione del modello è necessario considerare le due situazioni principali che possono verificarsi in un contesto applicativo simile, ovvero:

- E' meglio considerare il modello più preciso ma con un numero di FP elevato, supponendo che una paziente abbia un tumore benigno quando in realtà presenta la situazione opposta?
- E' meglio considerare un modello non molto preciso, che quindi commette errori, ma sono errori dovuti a un elevato numero di FN, che porta costi aggiuntivi ma non impatta pesantemente sulla salute delle persone?

Dall'analisi effettuata è emerso che tutti i modelli utilizzati forniscono degli ottimi risultati, soprattutto in termini di accuratezza. In ultima analisi però è necessario individuare quali tra questi risulti essere il migliore in base al dominio applicativo richiesto. Infatti, come è stato introdotto all'inizio del progetto, non basta che un modello sia preciso ma deve anche saper limitare gli errori relativi al fatto di ignorare un possibile caso di tumore maligno classificandolo come benigno. A questo proposito, il classificatore Naive Bayes è risultato essere il migliore tra quelli proposti: presenta un'accuratezza del 97.65% e un ridotto numero di FP rispetto al totale degli errori che ha commesso. Tale risultato è stato ottenuto considerando una politica di rimozione dei valori mancanti e non considerando l'attributo 'Mitoses' all'interno del dataset (anche tutti gli altri esperimenti sugli altri modelli hanno prodotto i migliori risultati per questa combinazione di operazioni sul dataset). Gli altri modelli invece, nonostante siano stati anch'essi molto precisi nella classificazione (accuratezza intorno al 95/96%), presentano alcuni problemi dovuti all'eccessivo numero di errori. In particolare, J48 presenta un elevato numero sia di FN che di FP, il che non lo rende del tutto affidabile per un compito così delicato. Stessa cosa per JRip che commette errori sia per quanto riguarda i FP che i FN, perciò è rischioso adottare un classificatore di questo genere in quanto c'è il rischio che non riconosca un tumore maligno classificandolo come benigno o viceversa.

5.1 Ulteriori Esperimenti

Per valutare ulteriormente la bontà dei modelli di classificazione, sono state confrontate le rispettive **AUC** (Area Under Curve).



(a) JRip (b) J48 (c) NaiveBayes

Figure 22: Confronto ROC Curve dei modelli.

Da questo confronto emerge che tutti i modelli hanno buona prestazioni di classificazione ($AUC = 0.95$ circa), ma il classificatore bayesiano è quello che meglio di tutti sembra svolgere correttamente la distinzione di tumori benigni da quelli maligni.
Infine, come ulteriore prova di correttezza che il classificatore bayesiano sia quello più performante, è stato calcolato il **lift** del modello e confrontato con quello dei due classificatori "sfidanti". Facendo riferimento alla situazione in cui sono stati rimossi i valori mancanti e l'attributo 'Mitoses' dal dataset, il lift prodotto dal modello bayesiano è $\frac{431/434}{444/683} = 1,5276$. Mentre per il modello che sfrutta l'albero decisionale si ha $\frac{423/436}{444/683} = 1,4924$ e per quello che utilizza le regole si ha $\frac{428/448}{444/683} = 1,4696$. Come risultato degli esperimenti precedenti, anche in questo caso il modello bayesiano sembra ottenere il migliore indice di lift, perciò risulta essere il migliore per questo tipo di attività.