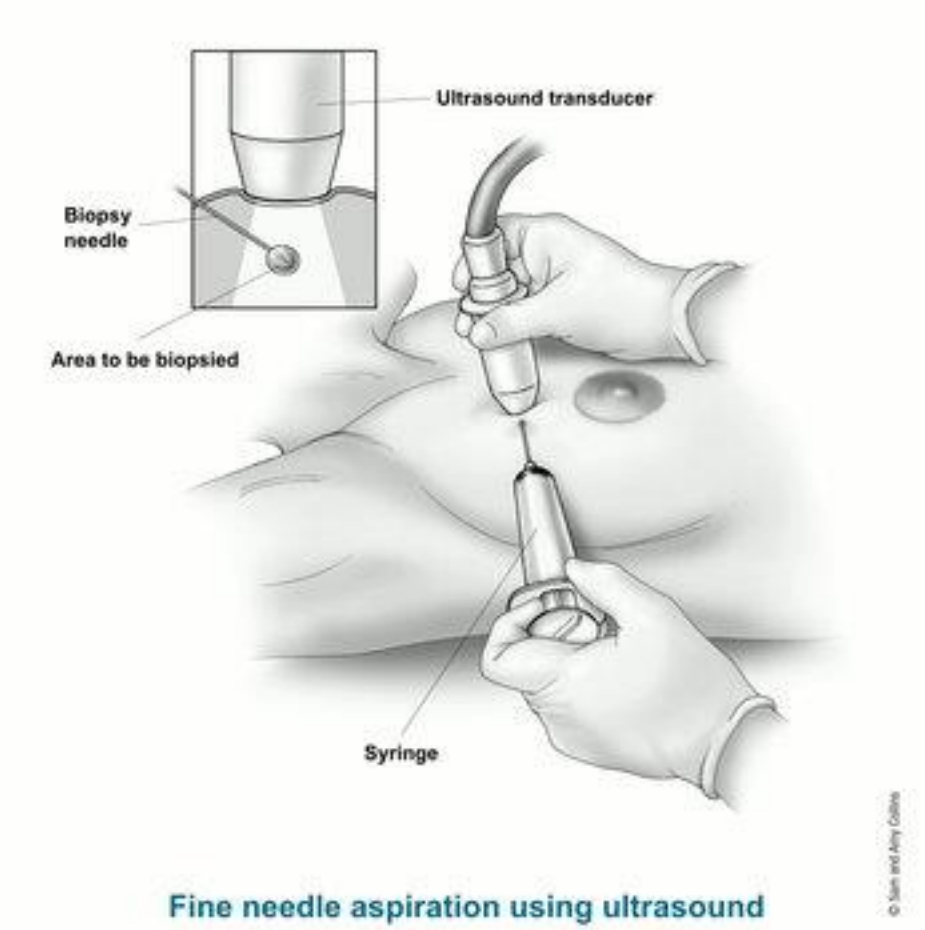
UNIVERSITÀ
DEGLI STUDI
FIRENZE

Breast Cancer Wisconsin

Progetto di Data Mining and Organization

a cura di Jacopo Manetti

Introduzione



Per questo progetto di data mining è stato preso il dataset **Breast Cancer Wisconsin** disponibile presso il sito dell'UCI, Il progetto che verrà descritto ha come obiettivo quello di costruire un modello di classificazione in grado di prevedere la diagnosi (maligno o benigno) di un tumore mammario a partire dalle caratteristiche delle cellule estratte da un'immagine digitale di un ago aspirato fine (FNA).

Data Understanding

Dataset	
1	ID
2	Diagnosis (M = maligno, B = benigno)
3	radius (media delle distanze dal centro)
4	texture (deviazione standard dei valori della scala di grigi)
5	perimeter
6	area
7	smoothness (variazione locale nelle lunghezze del raggio)
8	compactness (perimetro^2 / area - 1.0)
9	concavity (gravità delle porzioni concave del contorno)
10	concave points (numero di porzioni concave del contorno)
11	symmetry
12	fractal dimension ("approssimazione della linea di costa" - 1)

Questo dataset contiene informazioni su 569 campioni diagnosticati come benigni o maligni. Ciascun campione è stato analizzato 3 volte per calcolare 30 caratteristiche. Per ogni caratteristica è stato calcolato il valore medio, l'errore standard e il valore "peggiore" (il valore massimo tra i tre più grandi). Nel dataset sono presenti 357 campioni benigni e 212 maligni. Non sono presenti valori mancanti.

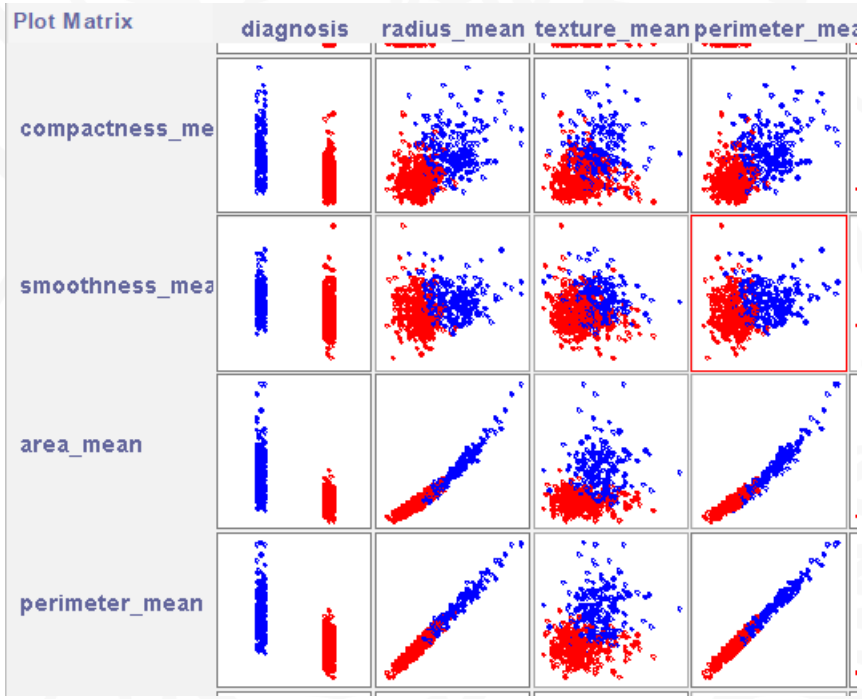
Il dataset si presenta già abbastanza pulito, non ci sono missing value e gli attributi sono tutti valori continui numerici ad eccezione di «diagnosis» che però è l'attributo classe.

Preprocessing

Per prima cosa viene eliminata la colonna id, che non è utile ai fini della classificazione, e si procede a una prima visualizzazione dei dati.

Nonostante gli attributi siano relativamente numerosi salta subito all'occhio che ci sia una correlazione tra gli attributi area, perimetro e raggio. L'eliminazione di attributi fortemente correlati tra loro, nota come selezione delle caratteristiche, può essere una buona idea poiché può aiutare a ridurre la dimensionalità dei dati e a migliorare l'accuratezza e la velocità degli algoritmi di classificazione.

In particolare, perimetro, raggio e area sono fortemente correlati tra loro poiché l'area è una funzione del raggio e del perimetro. Pertanto, l'eliminazione di uno di questi attributi può essere considerata senza compromettere la precisione del modello di classificazione.

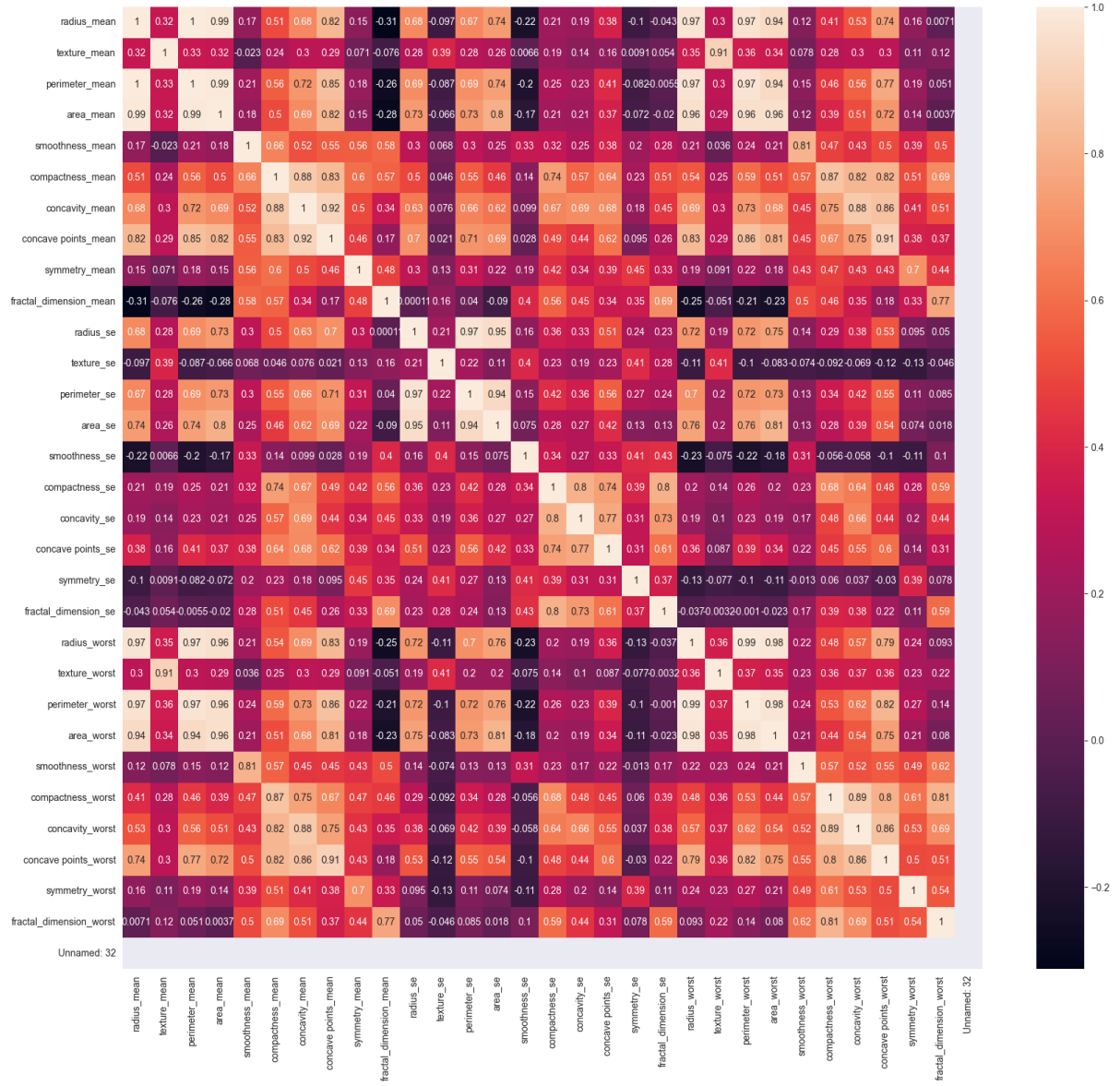


In figura una porzione della visualizzazione ottenibile con WEKA

UNIVERSITÀ DEGLI STUDI FIRENZE

Preprocessing - 2

Per visionare in maniera complessiva la correlazione tra gli attributi si ricorre ad un heatmap.

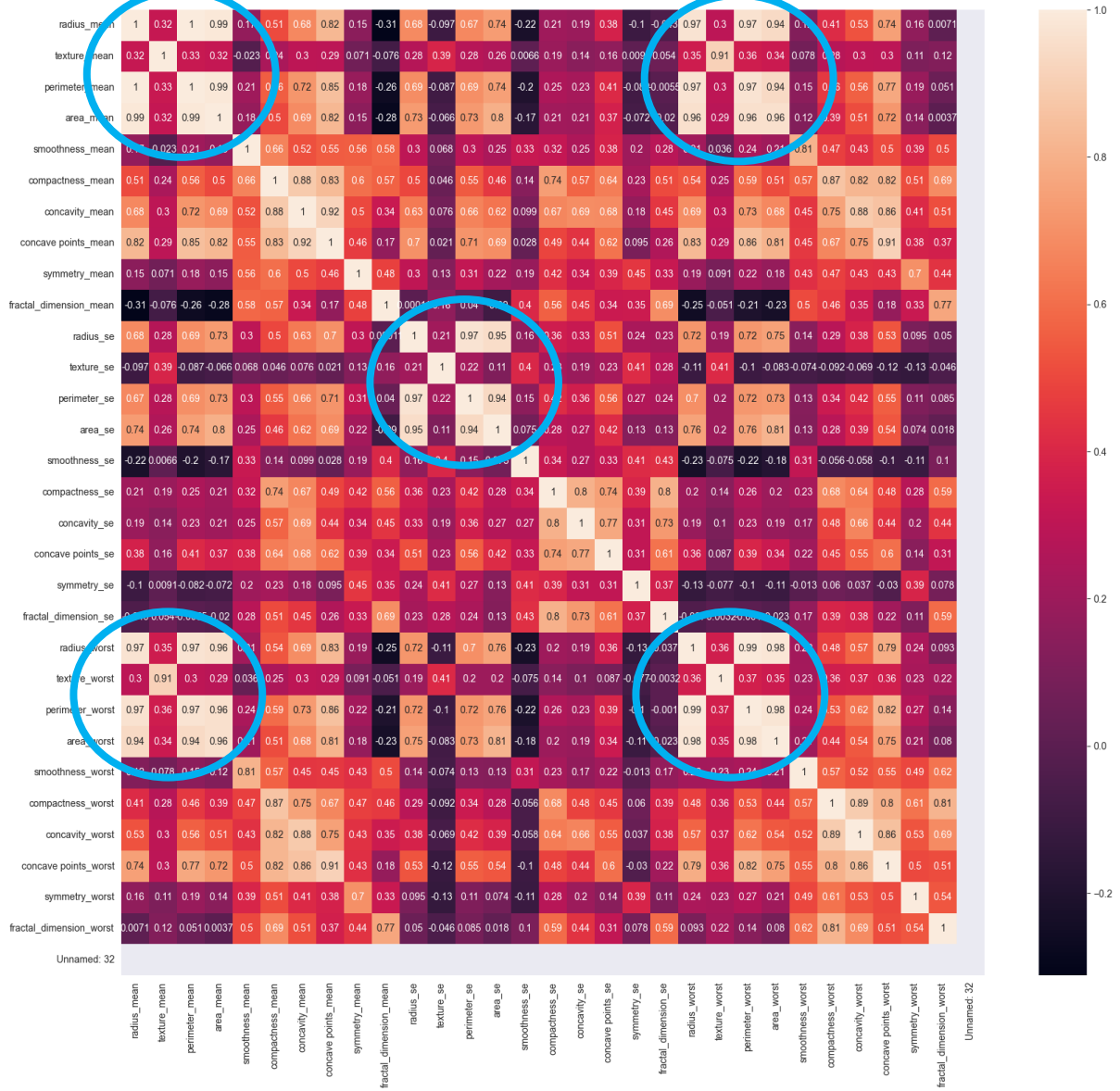


UNIVERSITÀ DEGLI STUDI FIRENZE

Preprocessing - 2

Per visionare in maniera complessiva la correlazione tra gli attributi si ricorre ad un heatmap.

Come previsto le zone con correlazione più alta sono quelle che riguardano area, perimetro e raggio.

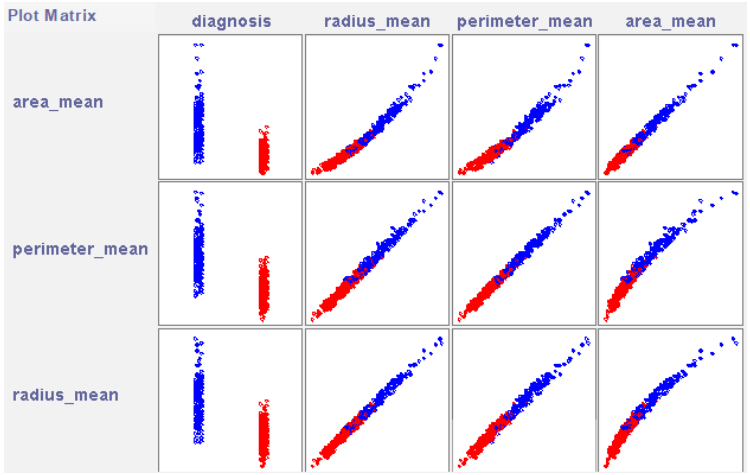


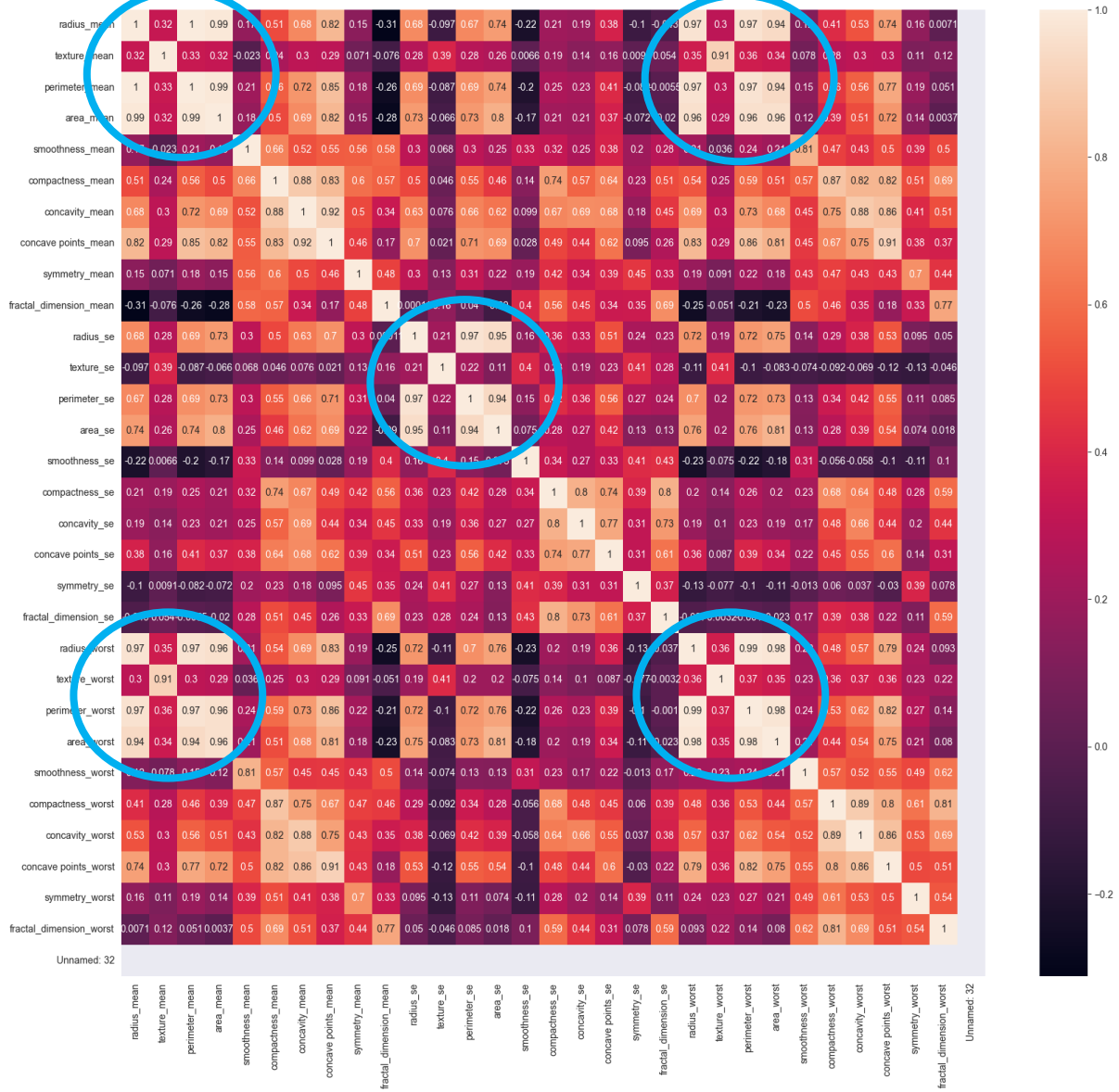
UNIVERSITÀ DEGLI STUDI FIRENZE

Preprocessing - 2

Per visionare in maniera complessiva la correlazione tra gli attributi si ricorre ad un heatmap.

Come previsto le zone con correlazione più alta sono quelle che riguardano area, perimetro e raggio.



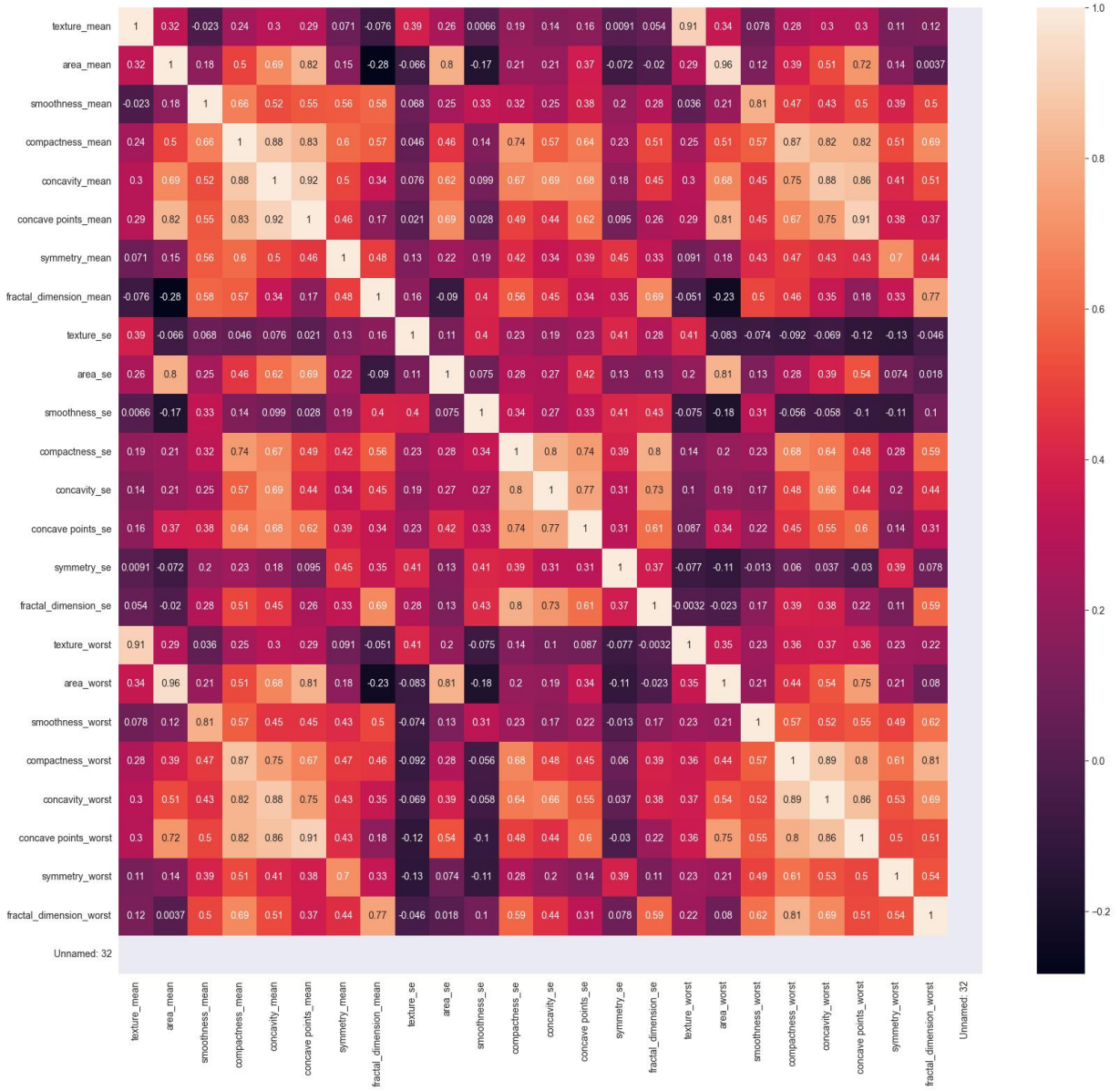


Decido quindi di eliminare gli attributi perimetro e raggio, lasciando solo l'area, dato che sembra il migliore nel separare benigno e maligno.

UNIVERSITÀ DEGLI STUDI FIRENZE

Preprocessing - 3

Ricalcolando l'heatmap ora otteniamo:

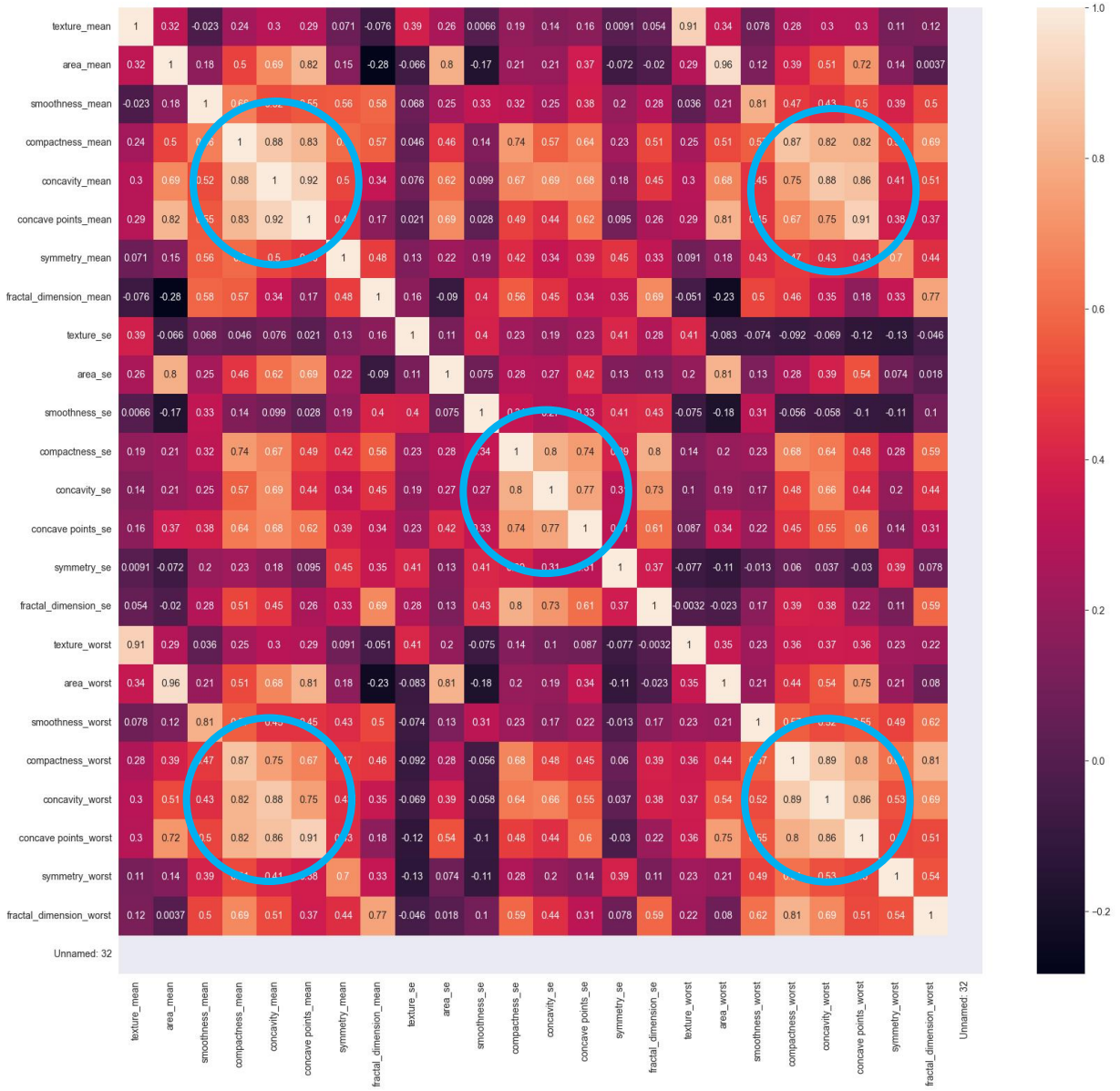


UNIVERSITÀ DEGLI STUDI FIRENZE

Preprocessing - 3

Ricalcolando l'heatmap ora otteniamo:

Ora si può notare con più facilità che anche gli attributi concavità, compattezza e concave points sono fortemente correlati tra loro.

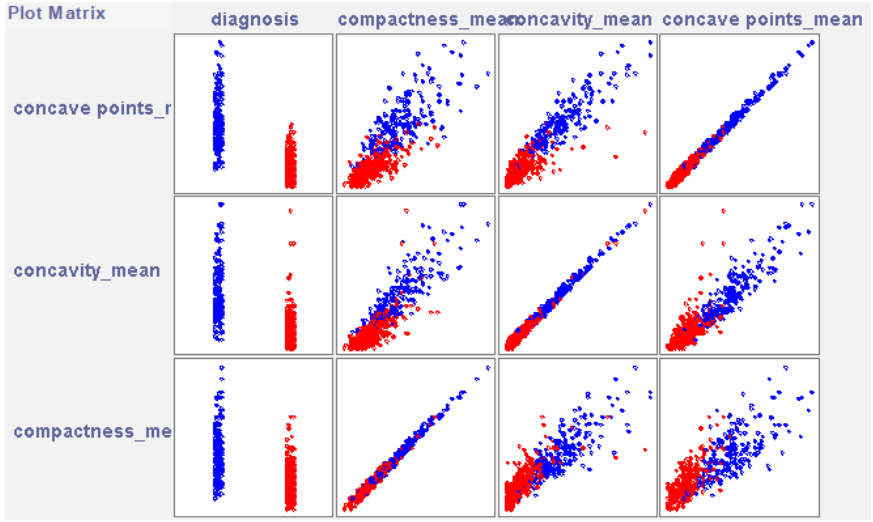


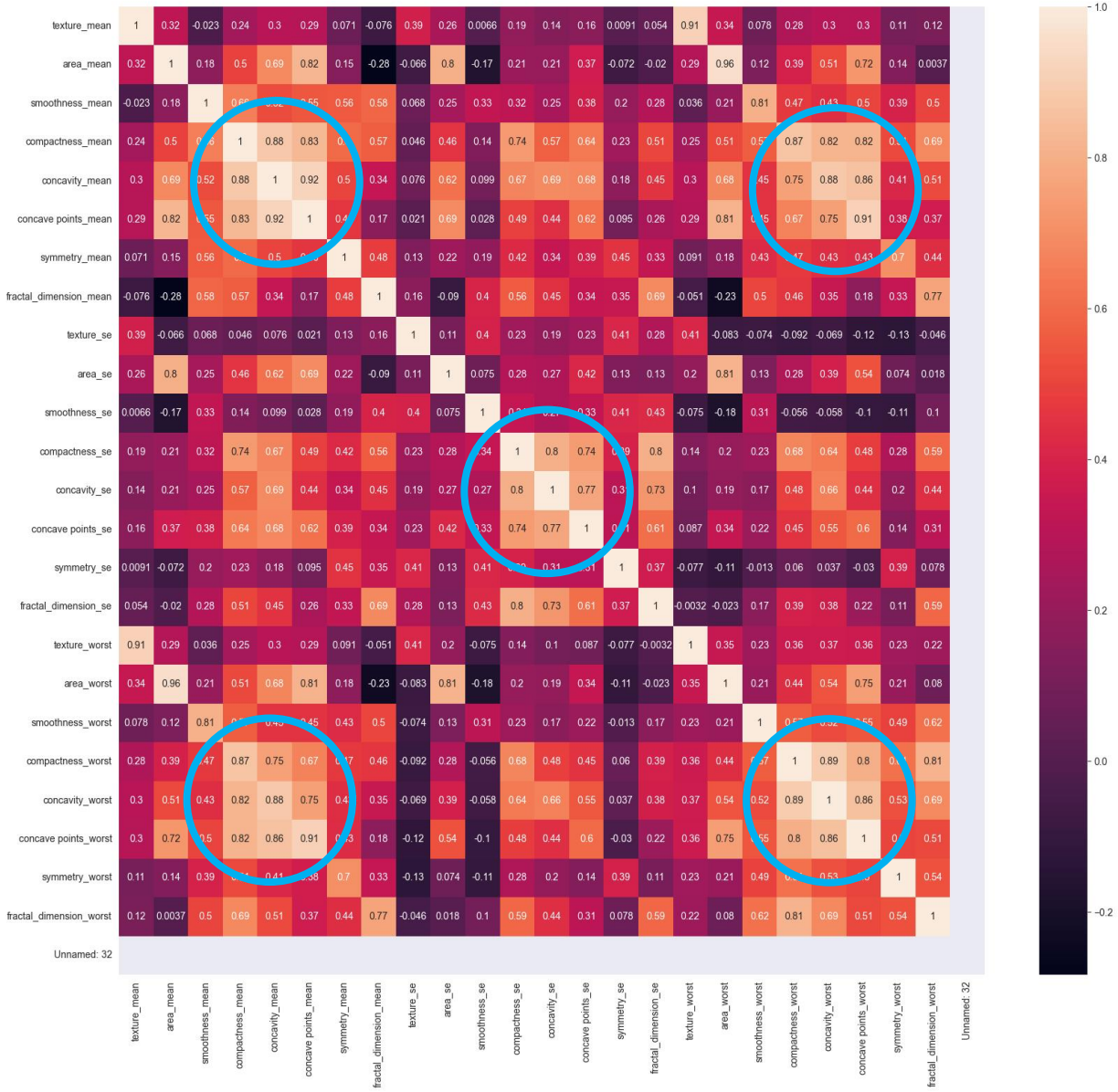
UNIVERSITÀ DEGLI STUDI FIRENZE

Preprocessing - 3

Ricalcolando l'heatmap ora otteniamo:

Ora si può notare con più facilità che anche gli attributi concavità, compattezza e concave points sono fortemente correlati tra loro.





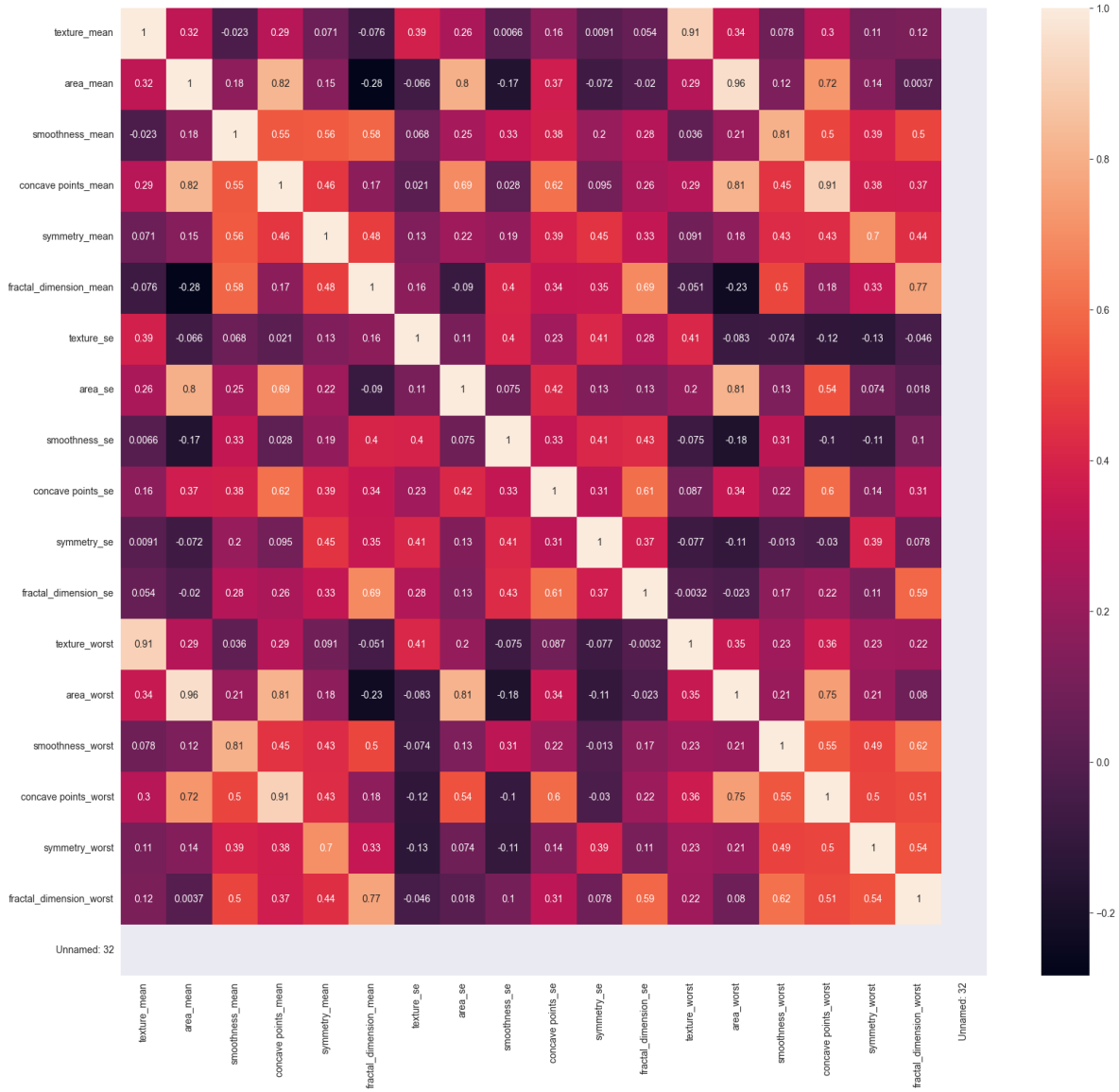
Decido quindi di eliminare gli attributi concavità e compattezza, lasciando solo concave points, dato che sembra il migliore nel separare benigno e maligno.

UNIVERSITÀ DEGLI STUDI FIRENZE

Preprocessing - 4

Ricalcolando l'heatmap ora otteniamo:

Infine, per avere una maggiore accuratezza e evitare problemi di scala, i dati vengono normalizzati.



Applicazione degli algoritmi

Per questo problema sono stati confrontati tra loro 4 algoritmi di data mining utili alla classificazione, utilizzando il software WEKA.

Algoritmi di classificazione			
K-nearest neighbors (IBk)	Naïve bayes	Albero di decisione C4.5 (j48)	Multylayer perceptron

K-nearest neighbors (IBk)

L'algoritmo k-nearest neighbors (k-NN) è un algoritmo di classificazione non parametrico che utilizza la distanza tra i k punti più vicini per classificare un nuovo punto.

In weka abbiamo k = 1 di default, Utilizzare k=1 per l'algoritmo k-NN significa che verrà considerato solo il punto di addestramento più vicino al nuovo punto da classificare. Questo può essere utile in alcuni casi, ad esempio quando si vuole classificare i dati in base a una relazione di prossimità molto stretta tra i punti. Tuttavia, utilizzare k=1 può anche essere pericoloso poiché un singolo outlier o un singolo punto di addestramento errato può influire significativamente sulla classificazione del nuovo punto.

Per questo utilizzare un valore più alto di k (ad esempio 5 o 9) può fornire una maggiore robustezza all'algoritmo.

Vengono quindi confrontati i risultati ottenuti utilizzando diversi valori di k per determinare quale valore fornisce i migliori risultati

K-nearest neighbors (IBk) - Risultati

Training set

K=1

```
=== Classifier model (full training set) ===

IB1 instance-based classifier
using 1 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.05 seconds

=== Summary ===

Correctly Classified Instances         569               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0.0018
Root mean squared error                  0.0018
Relative absolute error                  0.3745 %
Root relative squared error              0.3622 %
Total Number of Instances              569

=== Confusion Matrix ===

   a   b   <-- classified as
 212   0 |   a = M
   0 357 |   b = B
```

K=5

```
=== Classifier model (full training set) ===

IB1 instance-based classifier
using 5 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.06 seconds

=== Summary ===

Correctly Classified Instances         556               97.7153 %
Incorrectly Classified Instances        13                2.2847 %
Kappa statistic                          0.9506
Mean absolute error                      0.046
Root mean squared error                  0.1383
Relative absolute error                  9.8394 %
Root relative squared error             28.6042 %
Total Number of Instances              569

=== Confusion Matrix ===

   a   b   <-- classified as
 200  12 |   a = M
   1 356 |   b = B
```

K=9

```
=== Classifier model (full training set) ===

IB1 instance-based classifier
using 9 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.06 seconds

=== Summary ===

Correctly Classified Instances         550               96.6608 %
Incorrectly Classified Instances        19                3.3392 %
Kappa statistic                          0.9277
Mean absolute error                      0.0593
Root mean squared error                  0.1601
Relative absolute error                 12.6892 %
Root relative squared error             33.1103 %
Total Number of Instances              569

=== Confusion Matrix ===

   a   b   <-- classified as
 196  16 |   a = M
   3 354 |   b = B
```

Cross validation (10 fold)

K=1

```
=== Classifier model (full training set) ===

IB1 instance-based classifier
using 1 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         541               95.0791 %
Incorrectly Classified Instances        28                4.9209 %
Kappa statistic                          0.8939
Mean absolute error                      0.051
Root mean squared error                  0.2214
Relative absolute error                 10.8972 %
Root relative squared error             45.7923 %
Total Number of Instances              569

=== Confusion Matrix ===

   a   b   <-- classified as
 194  18 |   a = M
  10 347 |   b = B
```

K=5

```
=== Classifier model (full training set) ===

IB1 instance-based classifier
using 5 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         545               95.7821 %
Incorrectly Classified Instances        24                4.2179 %
Kappa statistic                          0.9087
Mean absolute error                      0.0647
Root mean squared error                  0.185
Relative absolute error                 13.8268 %
Root relative squared error             38.2664 %
Total Number of Instances              569

=== Confusion Matrix ===

   a   b   <-- classified as
 194  18 |   a = M
   6 351 |   b = B
```

K=9

```
=== Classifier model (full training set) ===

IB1 instance-based classifier
using 9 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         547               96.1336 %
Incorrectly Classified Instances        22                3.8664 %
Kappa statistic                          0.9163
Mean absolute error                      0.0722
Root mean squared error                  0.1833
Relative absolute error                 15.4473 %
Root relative squared error             37.9176 %
Total Number of Instances              569

=== Confusion Matrix ===

   a   b   <-- classified as
 195  17 |   a = M
   5 352 |   b = B
```

K-nearest neighbors (IBk) - Risultati

Training set

K=1

```
=== Classifier model (full training set) ===

IB1 instance-based classifier
using 1 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.05 seconds

=== Summary ===

Correctly Classified Instances         569               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0.0018
Root mean squared error                  0.0018
Relative absolute error                  0.3745 %
Root relative squared error              0.3622 %
Total Number of Instances              569

=== Confusion Matrix ===

   a   b   <-- classified as
 212   0 |   a = M
   0 357 |   b = B
```

K=5

```
=== Classifier model (full training set) ===

IB1 instance-based classifier
using 5 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.06 seconds

=== Summary ===

Correctly Classified Instances         556               97.7153 %
Incorrectly Classified Instances        13                2.2847 %
Kappa statistic                          0.9506
Mean absolute error                      0.046
Root mean squared error                  0.1383
Relative absolute error                  9.8394 %
Root relative squared error             28.6042 %
Total Number of Instances              569

=== Confusion Matrix ===

   a   b   <-- classified as
 194  18 |   a = M
   6 351 |   b = B
```

K=9

```
=== Classifier model (full training set) ===

IB1 instance-based classifier
using 9 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.06 seconds

=== Summary ===

Correctly Classified Instances         550               96.6608 %
Incorrectly Classified Instances        19                3.3392 %
Kappa statistic                          0.9277
Mean absolute error                      0.0593
Root mean squared error                  0.1601
Relative absolute error                 12.6892 %
Root relative squared error             33.1103 %
Total Number of Instances              569

=== Confusion Matrix ===

   a   b   <-- classified as
 195  17 |   a = M
   5 352 |   b = B
```

Cross validation
(10 fold)

K=

```
=== Classifier model (full training set) ===

IB1 instance-based classifier
using 1 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         541               95.0791 %
Incorrectly Classified Instances        28                4.9209 %
Kappa statistic                          0.8939
Mean absolute error                      0.051
Root mean squared error                  0.2214
Relative absolute error                 10.8972 %
Root relative squared error             45.7923 %
Total Number of Instances              569

=== Confusion Matrix ===

   a   b   <-- classified as
 194  18 |   a = M
  10 347 |   b = B
```

9

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         545               95.7821 %
Incorrectly Classified Instances        24                4.2179 %
Kappa statistic                          0.9087
Mean absolute error                      0.0647
Root mean squared error                  0.185
Relative absolute error                 13.8268 %
Root relative squared error             38.2664 %
Total Number of Instances              569

=== Confusion Matrix ===

   a   b   <-- classified as
 194  18 |   a = M
   6 351 |   b = B
```

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         547               96.1336 %
Incorrectly Classified Instances        22                3.8664 %
Kappa statistic                          0.9163
Mean absolute error                      0.0722
Root mean squared error                  0.1833
Relative absolute error                 15.4473 %
Root relative squared error             37.9176 %
Total Number of Instances              569

=== Confusion Matrix ===

   a   b   <-- classified as
 195  17 |   a = M
   5 352 |   b = B
```

Come previsto aumentando il valore di k il modello aumenta la sua robustezza

Naïve bayes

L'algoritmo di Naive Bayes è un algoritmo di classificazione basato sul Teorema di Bayes, che utilizza la probabilità per prevedere la classe di un nuovo punto.

La discretizzazione è utile con l'algoritmo di Naive Bayes perché esso funziona meglio con attributi categorici piuttosto che numerici. Quando si utilizzano attributi numerici, Naive Bayes deve calcolare le probabilità per ogni possibile valore dell'attributo, il che può diventare computazionalmente oneroso per grandi quantità di dati o per attributi con un gran numero di valori possibili.

Per fare questo, Weka permette al momento dell'applicazione dell'algoritmo di selezionare una funzione chiamata 'useSupervisedDiscretization' che, se impostata a true, utilizzerà un metodo di discretizzazione supervisionato per gli attributi numerici prima di utilizzare l'algoritmo di Naive Bayes, migliorando così le prestazioni dell'algoritmo perché gli attributi discretizzati hanno un minor numero di valori possibili e quindi meno probabilità di essere correlati tra loro.

UNIVERSITÀ DEGLI STUDI FIRENZE

Naïve bayes - Risultati

‘useSupervisedDiscretization’ = false

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         540               94.9033 %
Incorrectly Classified Instances        29                5.0967 %
Kappa statistic                          0.89
Mean absolute error                      0.0532
Root mean squared error                  0.2117
Relative absolute error                 11.3783 %
Root relative squared error             43.7859 %
Total Number of Instances              569

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,910     0,028     0,951       0,910   0,930      0,891    0,984     0,977     M
               0,972     0,090     0,948       0,972   0,960      0,891    0,984     0,979     B
Weighted Avg.  0,949     0,067     0,949       0,949   0,949      0,891    0,984     0,978

=== Confusion Matrix ===

   a   b   <-- classified as
 193  19 |   a = M
  10 347 |   b = B
```

‘useSupervisedDiscretization’ = true

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         545               95.7821 %
Incorrectly Classified Instances        24                4.2179 %
Kappa statistic                          0.9093
Mean absolute error                      0.0458
Root mean squared error                  0.1881
Relative absolute error                  9.8001 %
Root relative squared error             38.913  %
Total Number of Instances              569

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,929     0,025     0,956       0,929   0,943      0,909    0,989     0,986     M
               0,975     0,071     0,959       0,975   0,967      0,909    0,989     0,992     B
Weighted Avg.  0,958     0,054     0,958       0,958   0,958      0,909    0,989     0,990

=== Confusion Matrix ===

   a   b   <-- classified as
 197  15 |   a = M
   9 348 |   b = B
```

Naïve bayes - Risultati

‘useSupervisedDiscretization’ = false

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         540               94.9033 %
Incorrectly Classified Instances        29                5.0967 %
Kappa statistic                          0.89
Mean absolute error                      0.0532
Root mean squared error                  0.2117
Relative absolute error                 11.3783 %
Root relative squared error             43.7859 %
Total Number of Instances              569

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,910     0,028     0,951       0,910   0,930      0,891    0,984     0,977     M
               0,972     0,090     0,948       0,972   0,960      0,891    0,984     0,979     B
Weighted Avg.  0,949     0,067     0,949       0,949   0,949      0,891    0,984     0,978

=== Confusion Matrix ===

   a   b   <-- classified as
 193  19 |   a = M
  10 347 |   b = B
```

‘useSupervisedDiscretization’ = true

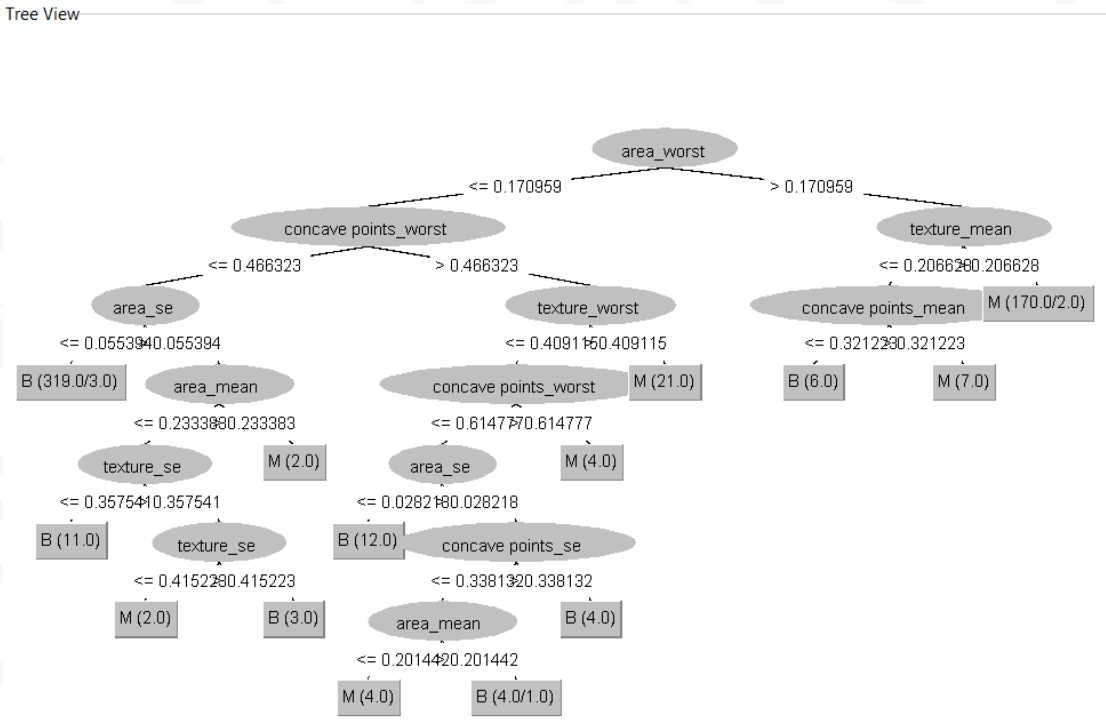
```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         545               95.7821 %
Incorrectly Classified Instances        24                4.2179 %
Kappa statistic                          0.9093
Mean absolute error                      0.0458

                                                                  easure  MCC      ROC Area  PRC Area  Class
               0,929     0,025     0,956       0,929   0,943      0,909    0,989     0,986     M
               0,975     0,071     0,959       0,975   0,967      0,909    0,989     0,992     B
Weighted Avg.  0,958     0,054     0,958       0,958   0,958      0,909    0,989     0,990

=== Confusion Matrix ===

   a   b   <-- classified as
 197  15 |   a = M
   9 348 |   b = B
```

Discretizzare ha migliorato l’accuratezza del modello

Albero di decisione C4.5 (j48)

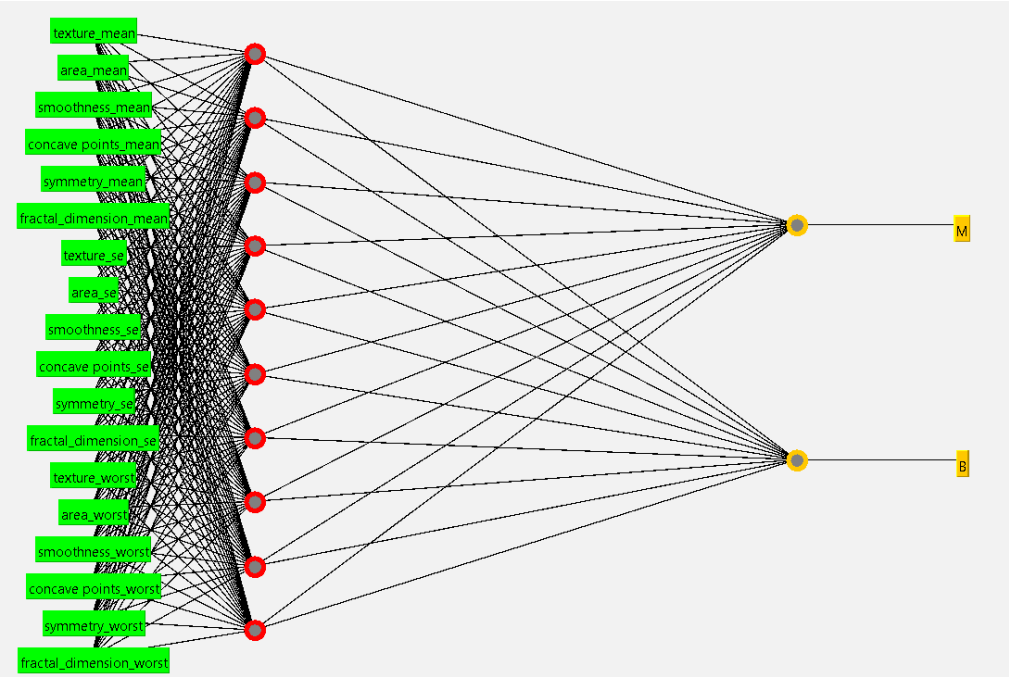
L'algoritmo C4.5 (noto anche come J48 in Weka) è un algoritmo di classificazione basato su decision tree che utilizza l'entropia per decidere come dividere i punti di addestramento in base alle loro classi.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         541               95.0791 %
Incorrectly Classified Instances        28                4.9209 %
Kappa statistic                          0.8947
Mean absolute error                      0.0593
Root mean squared error                  0.2213
Relative absolute error                 12.6899 %
Root relative squared error             45.7703 %
Total Number of Instances              569

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,934     0,039     0,934       0,934   0,934      0,895    0,926     0,869     M
               0,961     0,066     0,961       0,961   0,961      0,895    0,926     0,916     B
Weighted Avg.  0,951     0,056     0,951       0,951   0,951      0,895    0,926     0,898

=== Confusion Matrix ===

   a   b   <-- classified as
 198  14 |   a = M
  14 343 |   b = B
```



Multylayer perceptron

Il multilayer perceptron (MLP) è una tipologia di reti neurali artificiali che consiste in una serie di strati di neuroni interconnessi. In Weka, si può utilizzare l'algoritmo MultilayerPerceptron per utilizzare questo tipo di rete neurale. Di default, Weka utilizza un solo strato nascosto e normalizza il dataset su cui lavora, siccome questa operazione è già stata effettuata nel preprocessing questa seconda opzione viene impostata su false.




```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         549               96.4851 %
Incorrectly Classified Instances        20                3.5149 %
Kappa statistic                          0.9247
Mean absolute error                      0.0353
Root mean squared error                  0.1669
Relative absolute error                  7.5511 %
Root relative squared error             34.5196 %
Total Number of Instances              569

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,948     0,025     0,957      0,948     0,953      0,925    0,990     0,989     M
               0,975     0,052     0,969      0,975     0,972      0,925    0,990     0,992     B
Weighted Avg.  0,965     0,042     0,965      0,965     0,965      0,925    0,990     0,991

=== Confusion Matrix ===

   a   b   <-- classified as
 201  11 |   a = M
   9 348 |   b = B
```

Conclusione - Confronto dei risultati

Algoritmi di classificazione	Accuratezza del modello	Numero di falsi positivi
K-nearest neighbors (IBk) [k=9]	96,1336%	17
Naïve bayes	95,7821%	15
Albero di decisione C4.5 (j48)	95.0791%	14
Multylayer perceptron	96,4851%	11

Siccome tutti i modelli hanno ottenuto un'ottima accuratezza ho riportato anche il numero di falsi positivi, in quanto in questo caso specifico è importante che il numero di falsi positivi sia il più basso possibile, perché nella pratica è peggio lasciarsi sfuggire una cellula maligna classificandola come positiva.

In definitiva con il perceptron sono stati ottenuti i risultati migliori.

UNIVERSITÀ DEGLI STUDI FIRENZE

Fine