PROJET DATA SCIENCE

Breast Cancer Wisconsin

RESUME

Prédire le cancer du sein à partir des caractéristiques des noyaux cellulaires présent dans une image

Hazim Dahmani

Méthodes d'apprentissage

Table des matières

1. Motivation et positionnement du projet :

Le but du projet est de permettre un diagnostic du cancer du sein sans avoir à recourir à des méthodes et procédures de diagnostic coûteuses et chronophages.

La collecte des données du projet consiste tout simplement à calculer les caractéristiques des noyaux cellulaires à partir d'une image numérisée d'un aspirat à l'aiguille fine (FNA) d'une masse mammaire, La masse mammaire étant un échantillon.

Ainsi à travers ces caractéristiques, le projet est de prédire si cette masse cellulaire s'agisse d'une masse cellulaire bénigne ou maligne (cas du cancer)

Description des variables :

1) numéro d'identification

2) Diagnostic (M = malin, B = bénin)

Dix caractéristiques à valeur réelle sont calculées pour chaque noyau cellulaire:

1. Rayon (moyenne des distances du centre aux points du périmètre)
2. Texture (écart type des valeurs d'échelle de gris)
3. Périmètre
4. Zone
5. Lissé (variation locale des longueurs de rayon)
6. Compacité (périmètre ^ 2 / surface - 1,0)
7. Concavité (gravité des parties concaves du contour)
8. Points concaves (nombre de parties concaves du contour)
9. Symétrie
10. Dimension fractale ("approximation du littoral" - 1)

Plusieurs des articles énumérés ci-dessus contiennent des descriptions détaillées de comment ces caractéristiques sont calculées.

La moyenne, l'erreur standard et le "pire" ou le plus grand (moyenne des trois valeurs les plus élevées) de ces caractéristiques ont été calculées pour chaque image, résultant en 30 fonctionnalités. Par exemple, le champ 3 est le rayon moyen, le champ 13 est erreur standard de rayon, le champ 23 est pire rayon (correspondant à la moyenne des trois plus grandes valeurs des rayons de cellules dans chaque échantillon étant la masse mammaire).

Toutes les valeurs des caractéristiques sont recodées avec quatre chiffres significatifs.

2. Analyse descriptive :

Pour analyser la distribution des variables on peut procéder avec des outils analytique avec la commande summary affichant (min, max, mean, quartiles,median) ou graphique avec soit les histogrammes soit les boxplots.

a. Résultat analytique :

Voici un apperçu du resultat analytique :

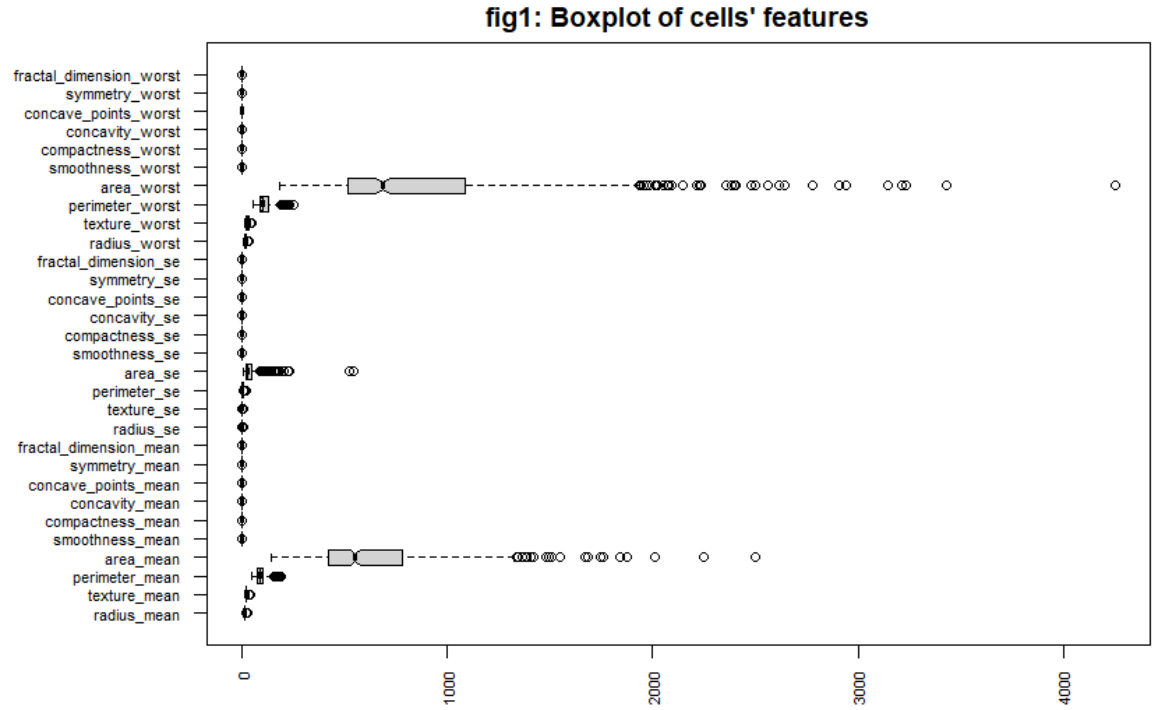
radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean
Min. : 6.981	Min. : 9.71	Min. : 43.79	Min. : 143.5	Min. :0.05263	Min. :0.01938
1st Qu.:11.700	1st Qu.:16.17	1st Qu.: 75.17	1st Qu.: 420.3	1st Qu.:0.08637	1st Qu.:0.06492
Median :13.370	Median :18.84	Median : 86.24	Median : 551.1	Median :0.09587	Median :0.09263
Mean :14.127	Mean :19.29	Mean : 91.97	Mean : 654.9	Mean :0.09636	Mean :0.10434
3rd Qu.:15.780	3rd Qu.:21.80	3rd Qu.:104.10	3rd Qu.: 782.7	3rd Qu.:0.10530	3rd Qu.:0.13040
Max. :28.110	Max. :39.28	Max. :188.50	Max. :2501.0	Max. :0.16340	Max. :0.34540

Déjà on peut remarquer que la data présente des échelles très différentes l'un de l'autre, par exemple ''area _mean'' a un max de 2501 et un min de 143,5 pourtant, ''compactness_mean'' a des valeurs qui ne dépasse même pas 0. Ce détail est très important quand on veut travailler avec des algorithmes sensibles à la distance. Ainsi une normalisation d'échelle peut s'avérer nécessaire (surtout dans la partie du non supervisé).

b. Résultat graphique :

On a opté pour le box plot afin de visualiser la différence d'échelle entre les variables, aini on remarque que ''area_mean'' et ''area_worst'' écrasent les boxplot des autres variables.

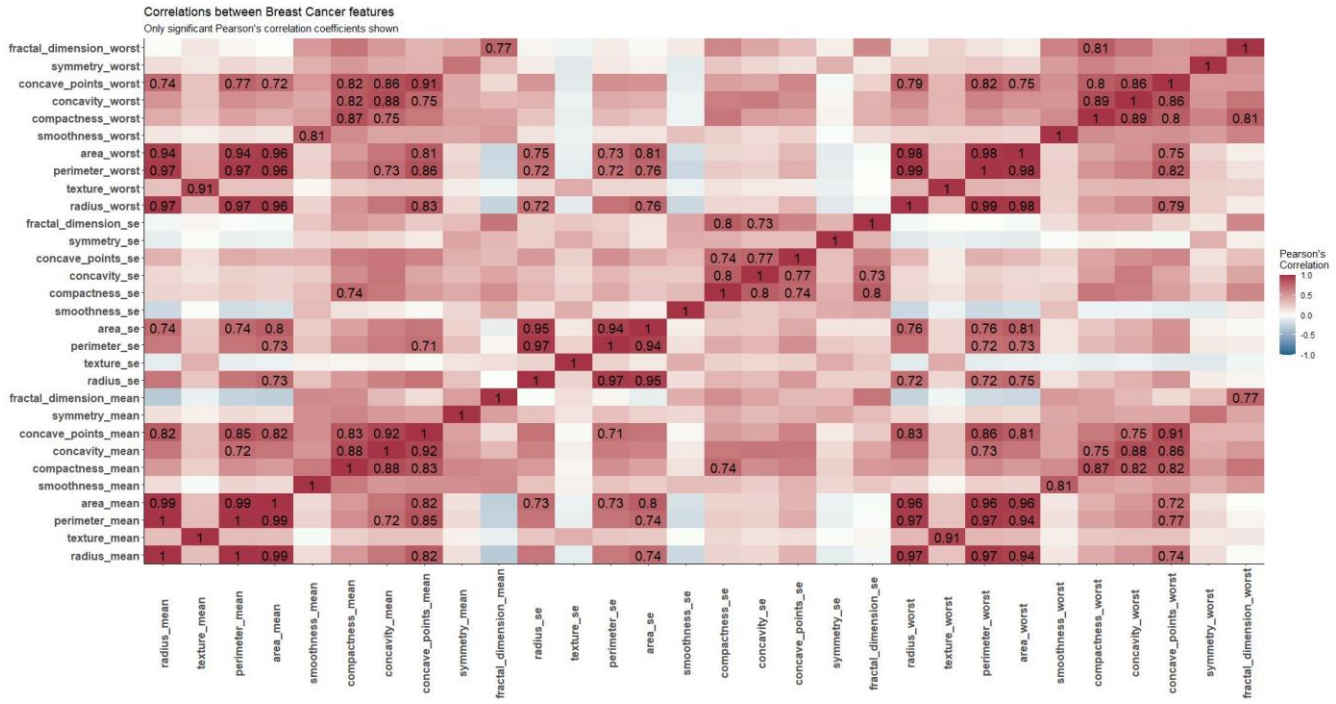
On remarque aussi la présence des outliers sur ces deux variables.



c. Interdépendance entre variables :

Pour illustrer l'interdépendance, on peut choisir le graphique généré par la commande ''plot(dataset)'' or avec autant de variables la visualisation sur le rapport ne sera pas possible, ainsi on vous invite de voir le graphe dans le dossier ''fig/Full-interdependency-plot-classfied.png''

Or la meilleure façon pour illustrer la dépendance linéaire entre les variables est de dessiner un heatmap :

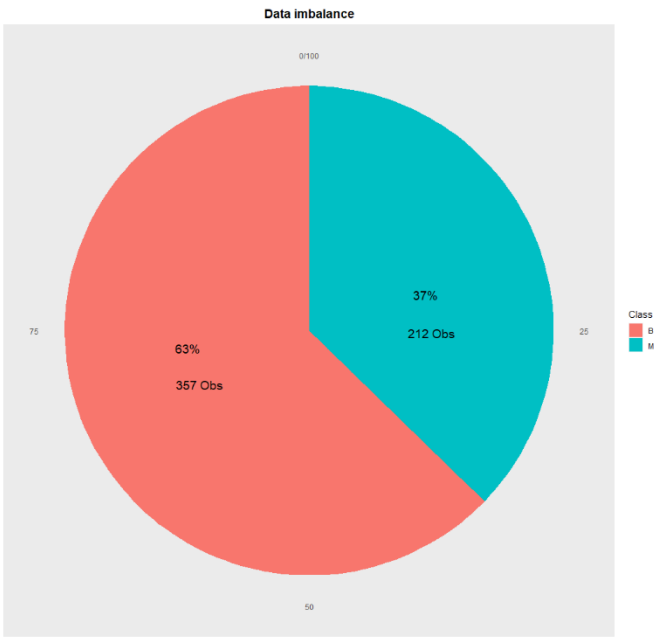


Les valeurs limites pour afficher le taux de corrélation (selon Pearson) sont 0,7 et -0,7 ainsi toute valeur hors l'intervalle $]-0{,}7,0{,}7[$ sera affiché.

D'après le Heatmap, on constate l'existence d'une dépendance linéaire entre plusieurs variables, quelques dépendances semblent naturelles/logiques telqu' entre ''radius_mean'' et ''perimeter_mean'', puisque la forme des cellules est approximativement circulaire (si on néglige l'épaisseur, puisque la source est une image), alors $perimiter = 2 \times \pi \times radius$ ainsi en appliquons la moyenne sur les deux parties de l'équation, on aura le justificatif de cette dépendance. Pour les autres variables, l'avis d'un expert du domaine peut être d'une assistance importante.

d. ''Data imbalance'' :

C'est important de vérifier si la dataset présente le problème de ''Data Imbalance'', à propos ''Data Imbalance'' est le fait quand dans les données pour une certaine out quelques classes dominent les autres classes, ainsi le modèle d'apprentissage produit souvent une structure déséquilibrée des erreurs, bien supérieur pour la classe minimale donnant en même temps un sens de précision élevé mais ''erronés''.



Pour notre cas il n'y a pas de problème de ''Data Imbalance'', sinon on devait procéder au ''Downsampling'' ou ''Upsampling''

3. Classification non supervisée :

Dans cette partie on va appliquer 3 algorithmes d'apprentissage non supervisé, précisément le clustering (Kmeans, PAM, CAH ou ''Hierarchical Clustering'').

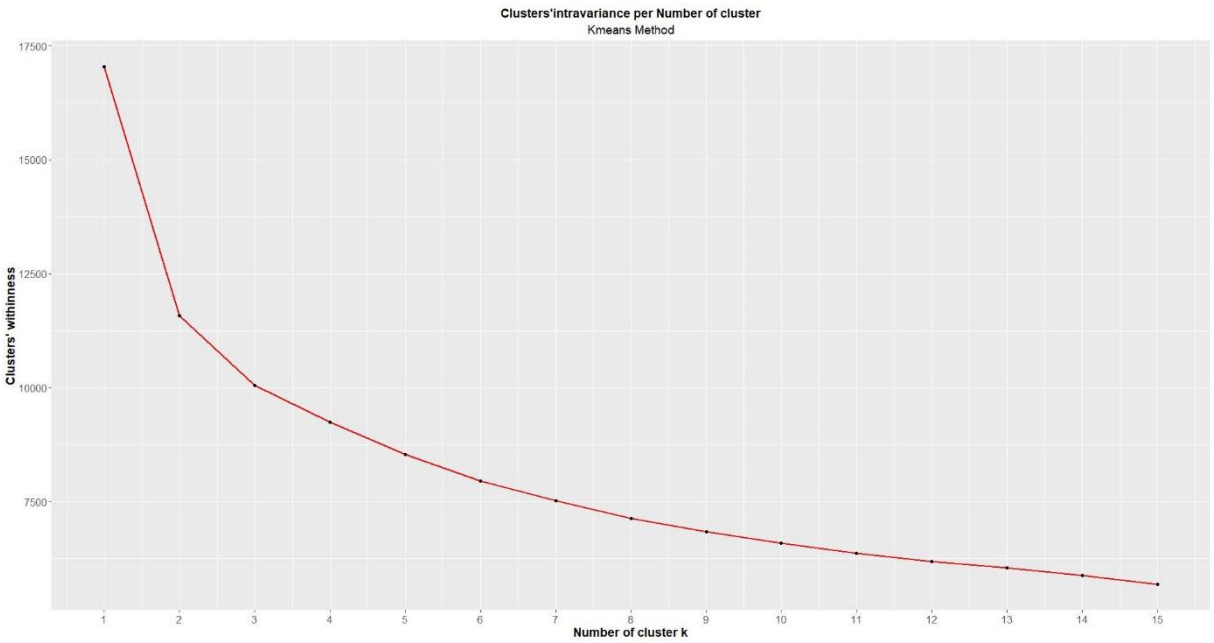
Le clustering établi par ces méthodes repose essentiellement sur des mesures de dissimilarité notamment la distance (souvent euclidienne), ainsi vu que notre Dataset présente des variables avec un échelle écrasant ceux des autres (voir le boxplot de 2.b). ainsi une normalisation est nécessaire afin d'équilibrer l'impacte des variables sur le clustering.

A. Kmeans :

a. Nombre de classes optimales

Pour déterminer le nombre optimal de clusters, on peut procéder à générer le graphes des intravariance par nombre de clusters et trouver le coude, le point au-delà du quelle on aura pas assez amélioration ou de diminution de cette intravariance, autrement dit, c'est juste de la ''complexité'' rajouté au clustering (ou un clustering forcé).

Le graphe suivant illustre le changement de l'intravariance globale au sein dans les clusters par nombre de cluster.



On peut remarquer un coude bien apparent au niveau du 2ème cluster, Or faire un partitionnement (''splitting'') en 3 ou 4 reste aussi une option envisageable.

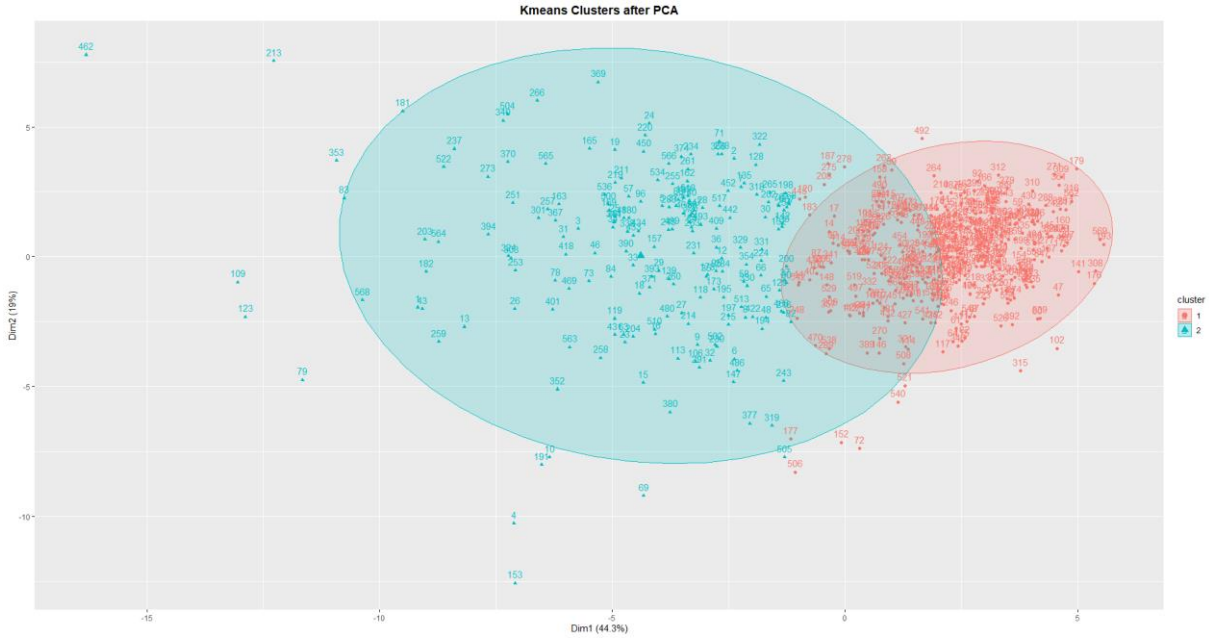
On va prendre dans ce cas 2 comme étant le nombre optimal de clusters pour le reste de l'analyse dans la méthode de Kmeans.

b. Extraction et interprétation :

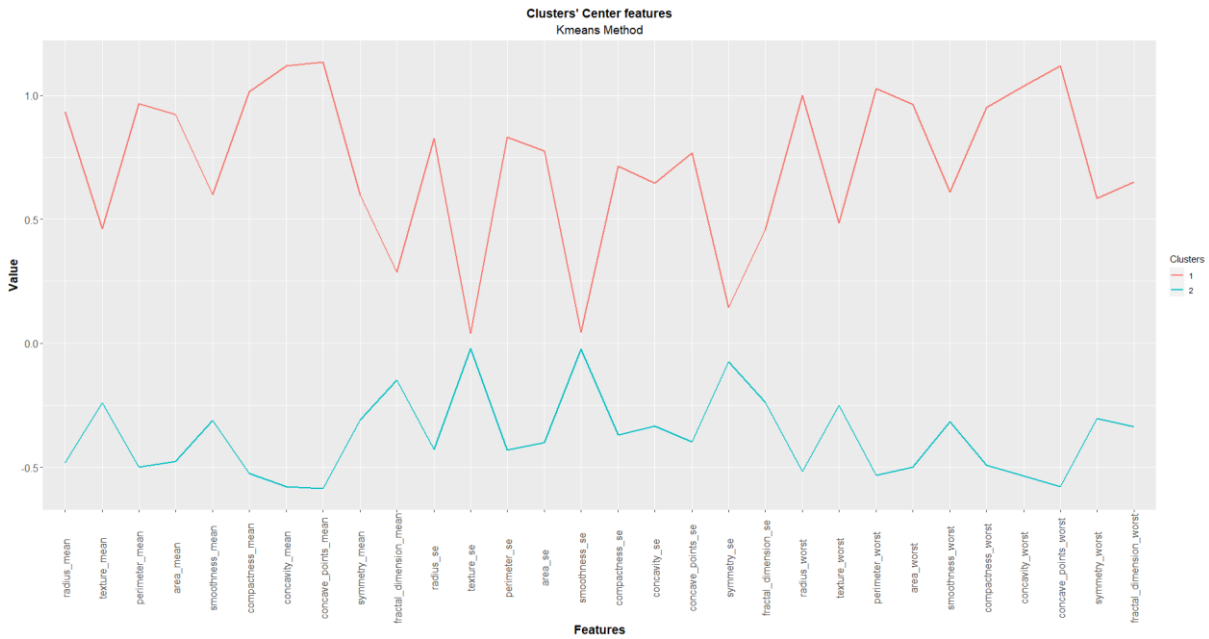
Pour permettre la visualisation de donnée et du clustering dans une dimension de 30 n'est pas possible, ainsi pour simplifier la visualisation qui s'approche le plus à la représentation de l'espace, on changera le référentiel et on projettera sur les deux axes qui expliqueront le maximum de la variance dans la data, un autre terme pour dire qu'on va appliquer la ''PCA'' (''Principal Components

Analysis''), on va affecter les points à leurs clusters correspondants. Pour faire tout cela, la librairie ''factoextra'' permet de faire tout cela en peu de commandes.

Le graphe suivant est un graphe généré grâce à cette librairie. Le clustering ''Kmeans'' est projeté sur 2 dimensions expliquant chacun respectivement 44,3% et 19% de la variance des points du dataset



Pour comparer les deux clusters, on peut comparer les centres des deux clusters, assumant que ces deux centres sont des bons représentants des clusters, ainsi le graphe suivant est une comparaison entre les variables des centres des deux clusters.



Dans le graphe ci-dessus, montre que les deux clusters ont un gap pour toutes les variables.

Certaines variables présentent un décalage très significatif entre les deux clusters telle que : ''concave_points_mean'', ''concave_points_worst'', ''concavity_mean''.

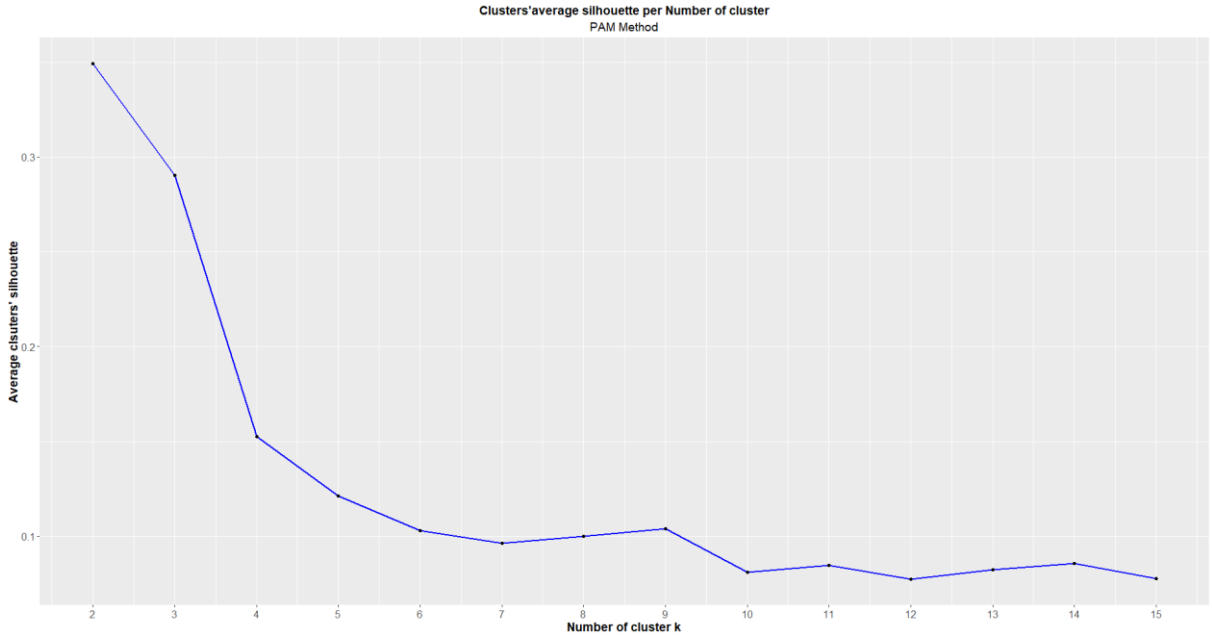
B. PAM :

PAM est une méthode de clustering puissante puisqu'elle repose sur un représentant luis donnant, la capacité de travailler avec plusieurs types de variables. Dans notre cas, la dataset est numérique, ainsi on va tirer plus avantage de ''silhouette'' comme étant un indicateur de performance très puissant et bien personnalisé pour chaque point de la dataset.

a. Nombre de classes optimales :

Pour déterminer le nombre de classes optimales dans le cas de ''PAM'', on peut prendre la moyenne de silhouette. A propos la silhouette, est le taux d'appartenance (en termes de distance) à la classe affectée.

Le graphe suivant illustre la silhouette moyenne par nombre de cluster.



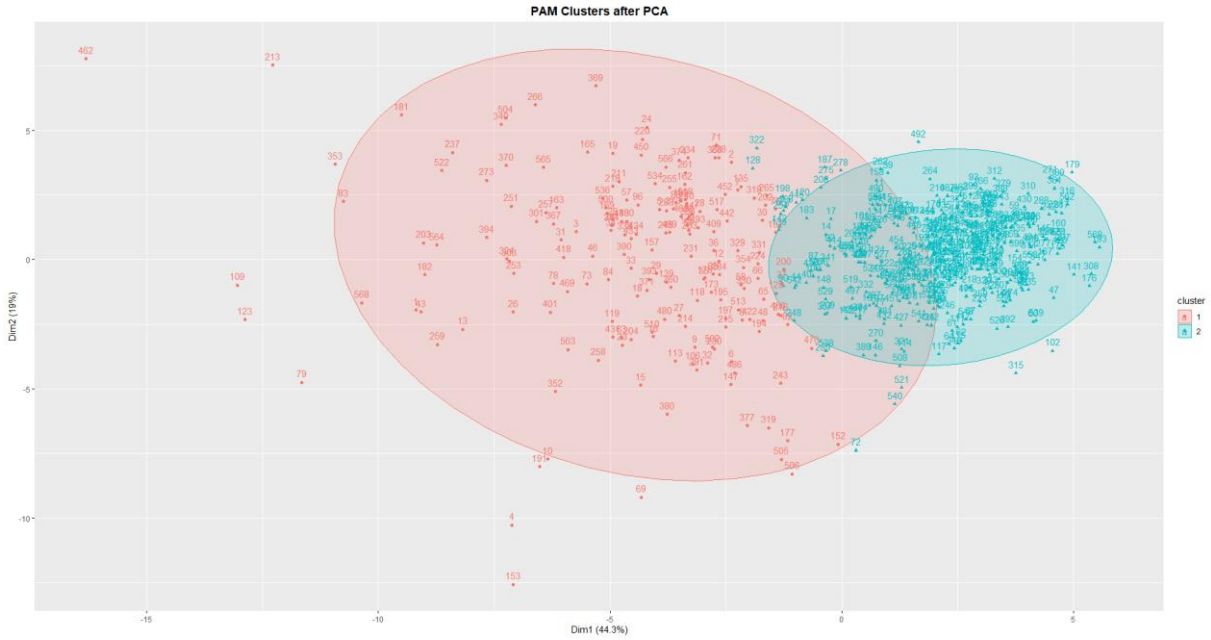
D'après le graphe ci-dessus, le partitionnement offrant la meilleure séparation entre les clusters (silhouette moyenne) est un partitionnement en deux clusters.

Il faut aussi mentionner que 0.35 comme étant la moyenne silhouette du clustering PAM n'est une bonne valeur pour la séparation indiquant un chevauchement entre les clusters dans le ''best case scenario''

b. Extraction et interprétation :

Pour la visualisation des deux clusters, on va procéder de la même manière que dans le Kmeans, on va utiliser la librairie ''factoextra'', exécutant le ''PCA'' pour projeter sur les deux dimensions expliquant le plus de variance.

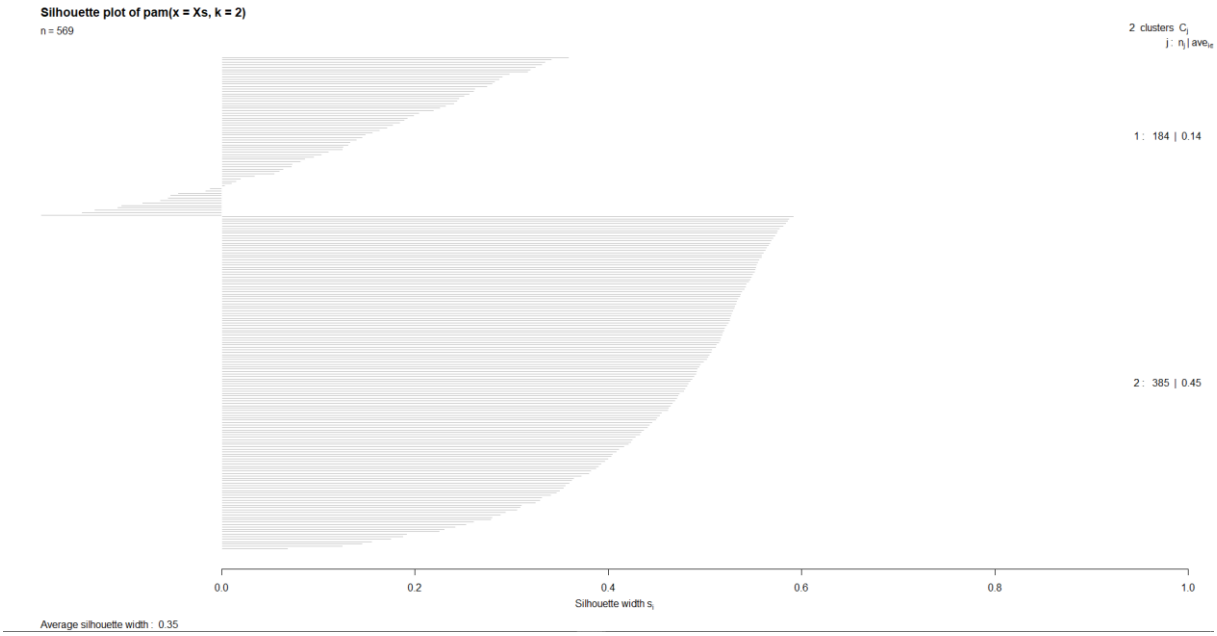
Le graphe suivant est un graphe généré grâce à cette librairie. Le clustering ‘’PAM’’ est projeté sur 2 dimensions expliquant chacun respectivement 44,3% et 19% de la variance des points du dataset.



D’après le groupe ci-dessus, on peut déjà voir un chevauchement entre les deux clusters expliquant la basse valeur moyenne de silhouette.

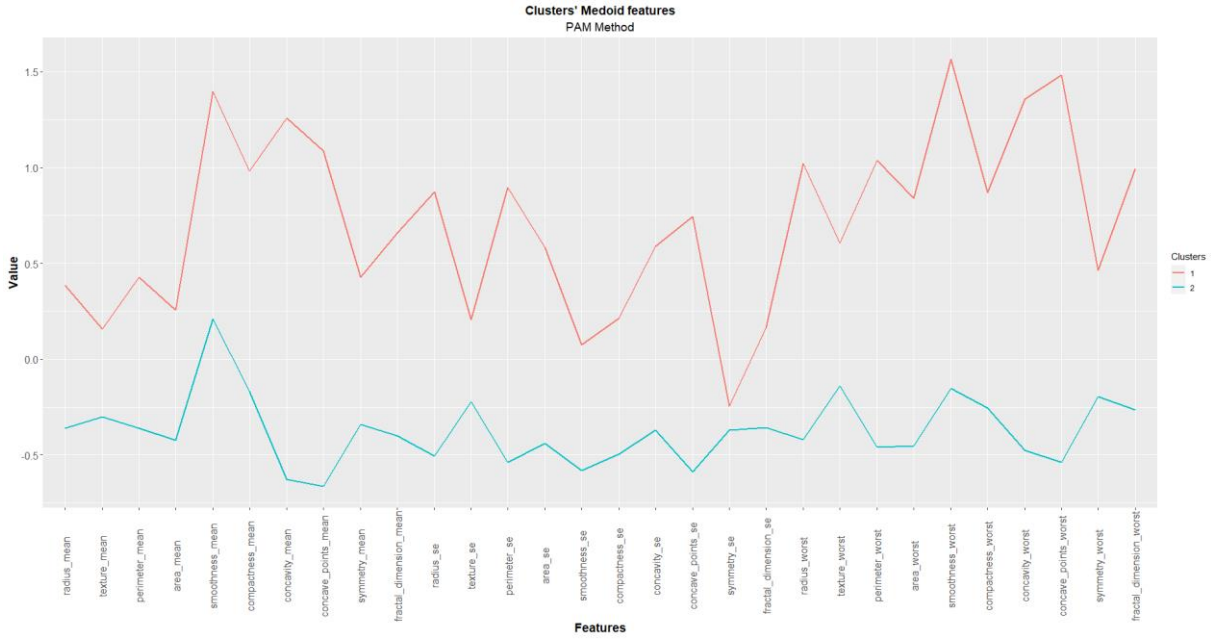
Le graphe ci-dessous montre la silhouette de chaque point, la silhouette moyenne de chaque cluster, les deux clusters présensent une valeur de silhouettre moyenne relativement très basse, indiquant une qualité basse de séparation des deux clusters l’un de l’autre.

Quelque points présentent des silhouettes négatives ces points sont à être affecté dans l’autre cluster.



Pour l'interprétation des deux clusters, on peut s'appuyer sur les points représentant chaque cluster et comparer plutôt leurs variables.

Le graphe suivant est une illustration des valeurs normalisées des représentants des deux clusters.



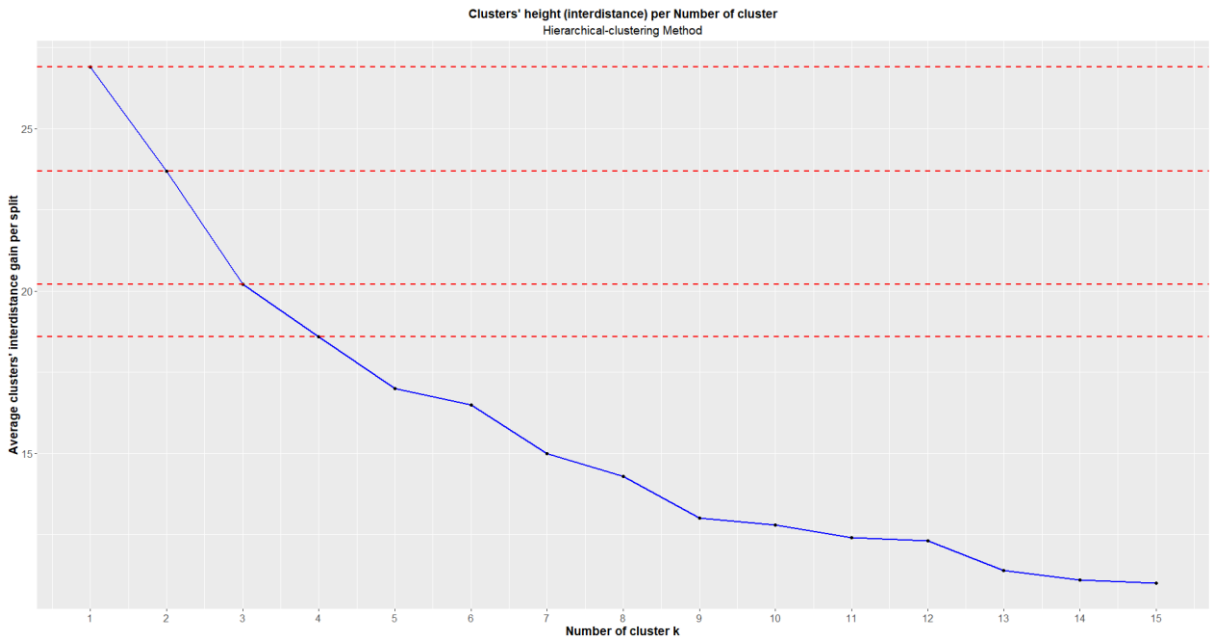
D'après le graphe ci-dessus, on peut voir des différences entre les deux clusters présentant des décalages sur toutes les dimensions (variables).

Certaines variables présentent un décalage très significatif entre les deux clusters telle que : ''smoothness_worst'', ''concave_points_worst'', ''concavity_worst'' et ''smoothness mean''.

C. CAH :

a. Nombre de classes optimales

Pour déterminer le nombre optimal de clusters, on a procédé en s'inspirant du concept du coude appliqué dans le Kmeans, à appliquer ce concept à l'interdistance entre les clusters dans CAH.

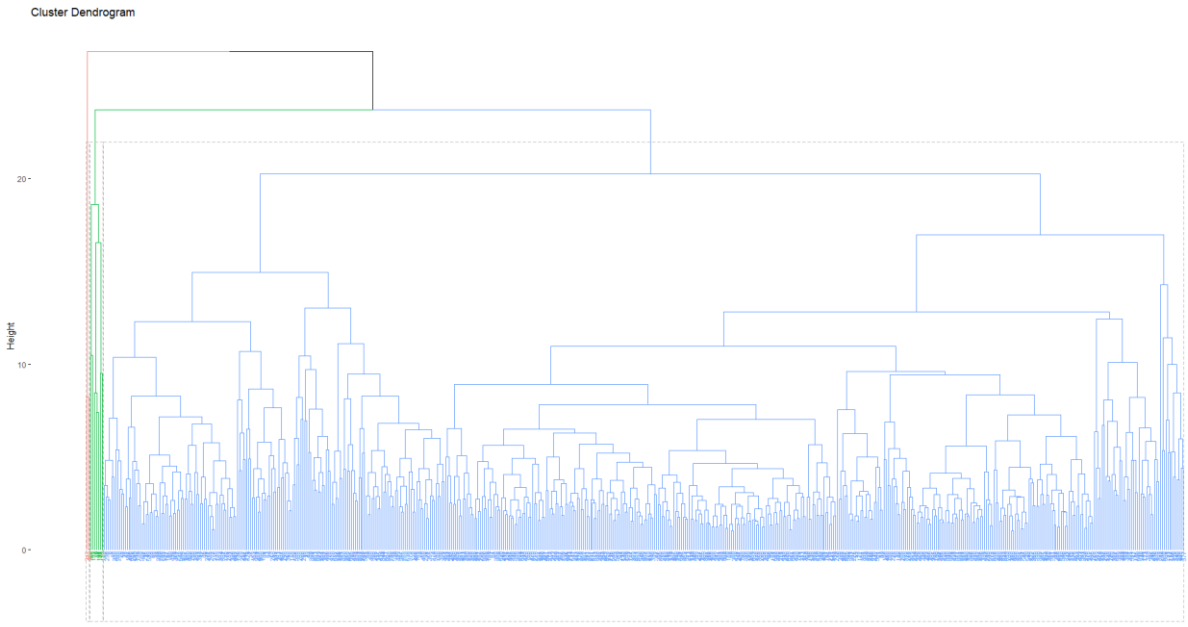


Le graphe ci-dessus représente l'interdistance entre les clusters par nombre de clusters.

D'après le graphe, 3 et 4 sont deux candidates au partitionnement optimal des points du dataset. On prendra 3 comme étant le nombre de cluster optimal dans le restant de l'étude avec la méthode CAH.

b. Extraction et interprétation :

Voici le dendrogramme avec un partitionnement visuelle avec les couleurs des 3 clusters existant dans nos données



D'après le graphe, on peut voir 3 clusters, celui orange est un cluster isolé qui comprends peut de point, un autre cluster (vert) plus proche du bleu qui comprend peu de points comparés au bleu.

Pourtant puisque le cluster en bleu comporte plusieurs points qu'on peut aussi partitionner en deux ou 3 clusters. Cette visualisation est l'un des grands avantages du clustering.

4. Classification supervisée :

Dans cette partie, on travaillera principalement avec les arbres de décisions, or dans la sous partie d'évaluation de performance, on va comparer entre l'arbre de décision et les forêts (''Random Forest'')

A. Bootstrap et spécifications :

''Bootstrap'' est premièrement utilisé dans les statistiques afin d'estimer des indicateurs statistiques telle que la moyenne, médiane surtout vu la difficulté de l'accès à la population.

Dans notre cas, on utilisera cette technique pour estimer la performance prédictive de nos modèles statistiques.
Ainsi on va générer 100 échantillon ayant une taille de 70% de celle de notre dataset et on va tester pour le reste de chaque échantillon (les points qui ne sont pas échantillonnés).

B. Performance du ''Decision Tree'' :

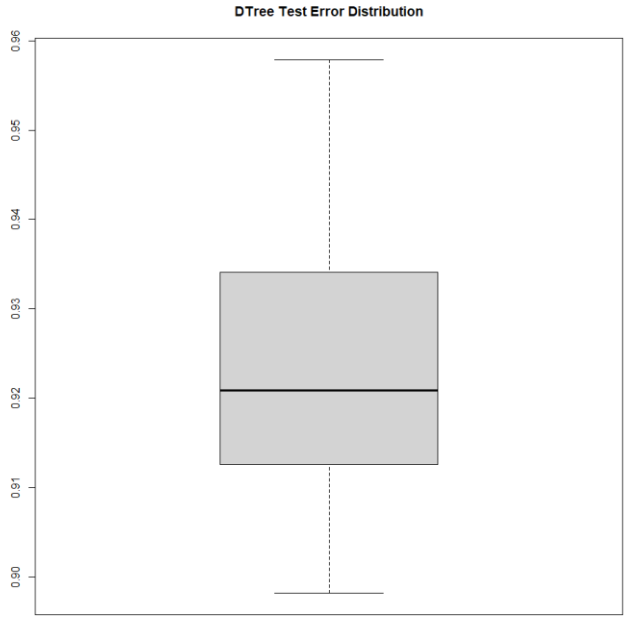
Pour évaluer la performance du ''Décision Tree'' à la classification dans ce cas, on va voir la consistance de cette performance à travers toute les ''bootstrapped datasets'' et on va la comparer à

celle du ‘’Random Forest’’ (une version avancée et plus stable de ‘’Decision Tree’’, à travers la méthode d’Ensemble ‘’Bagging’’ appliqué à plusieurs arbre sur des ‘’bootstrapped datasets or samples’’)

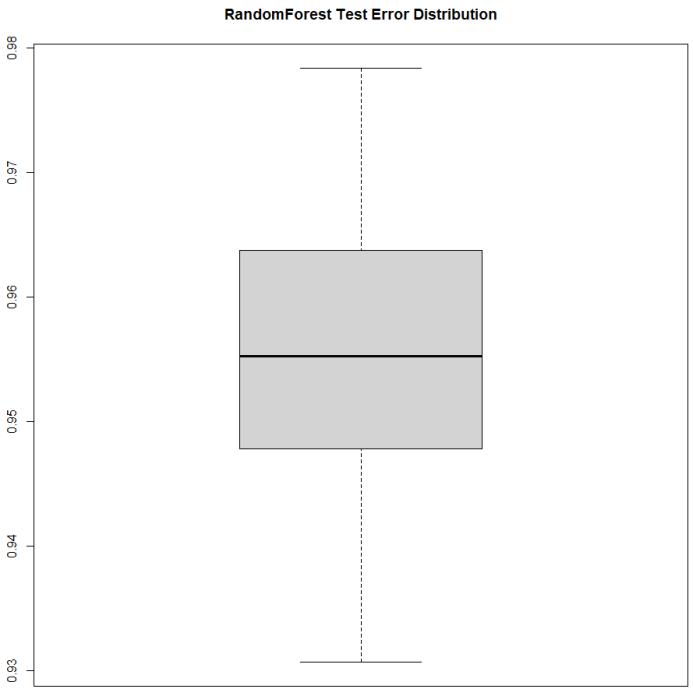
a. Distribution de l’erreur test (Decision Tree vs Random Forest) :

Après avoir appliquer le bootstrapping et la génération d’un vecteur d’erreur pour les arbres de décision et les forêts, on a résumer la distribution d’erreur à travers des box plot pour chaque méthode.

Le graphe ci-dessous représente la distribution d’erreur de ‘’Decision Tree’’ sur les 100 échantillons générés à travers le ‘’bootstrapping’’.
On peut remarquer qu’il n’y a pas de valeurs aberrantes, la précision moyenne est 92,4% (à partir du script R pas du graphe) avec une précision minimal proche de 90% ce qui est relativement une bonne performance.



Le graphe représente la distribution d’erreur de ‘Random Forest’’ sur les 100 échantillons générés à travers le ‘’bootstrapping’’.
On peut remarquer qu’il n’y a pas de valeurs aberrantes, la précision moyenne est 95,6% (à partir du script R pas du graphe) avec une précision dépassant un peu 93% ce qui est relativement une très bonne performance, Une bien plus supérieure à celle de ‘’Decision Tree’’ ce qui attendu et avec une variance bien plus inférieure, celle du ‘’Decision Tree’’ est 0,0032 alors que celle de ‘’Random Forest’’ est 0,0013, indiquant plus de consistance (‘’consistency or variability’’) dans la performance.



b. Matrix de confusion

La table ci-dessous représente la classification proposée par l'algorithme ''Decision Tree'' sur un échantillon test, ainsi on a 165 masses de cellules mammaires actuellement bénigne qui sont prédit comme étant bénigne et il y a 5 masses de cellules mammaires actuellement bénigne qui sont prédit comme étant maligne.

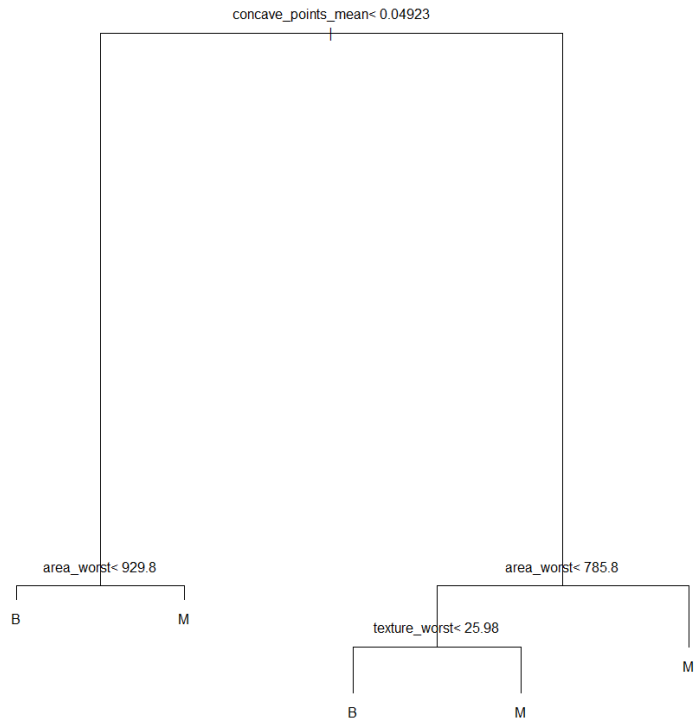
	B	M
B	167	5
M	8	100

La table ci-dessous représente la classification proposée par l'algorithme ''Random Forest'' sur un échantillon test.

	B	M
B	185	0
M	0	113

c. Interprétation et principales règles de decision :

Ci-dessous est un arbre décision le cas de notre dataset (''training partition'').



Ce script ci-dessous est une description de l'arbre en haut. Avec l'indentation marquant le niveau de partitionnement. Le tuple à côté montre le pourcentage de chaque classe après le partitionnement.

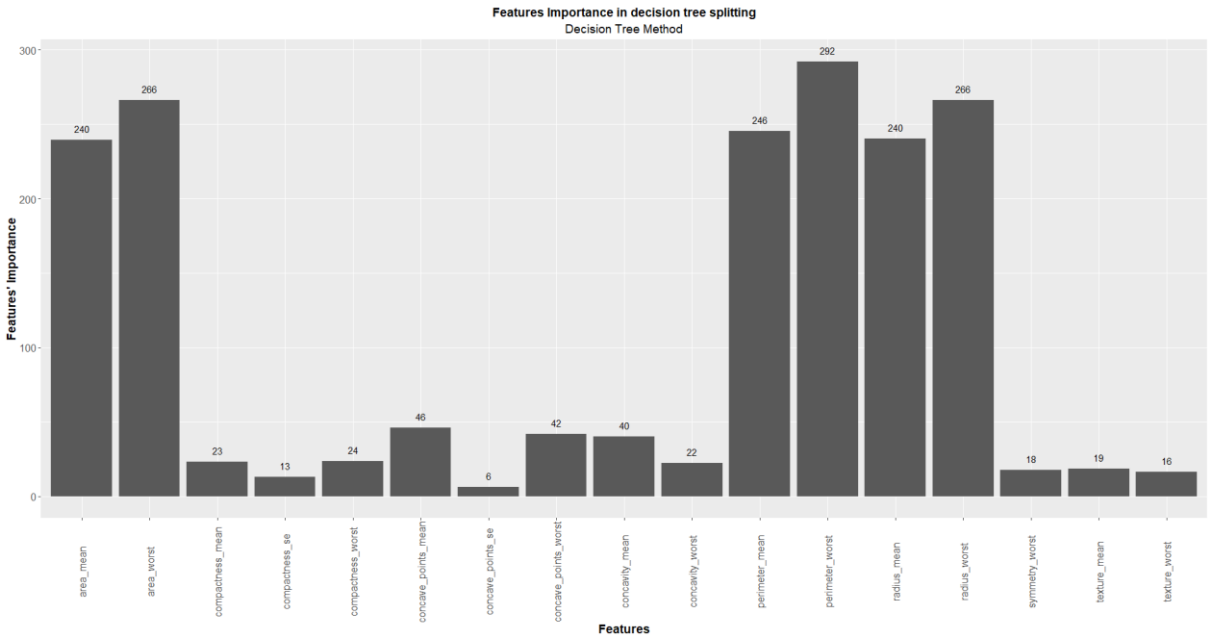
```
n= 397

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 397 148 B (0.627204030 0.372795970)
   2) concave_points_mean< 0.04923 247  13 B (0.947368421 0.052631579)
     4) area_worst< 929.8 234   5 B (0.978632479 0.021367521) *
     5) area_worst>=929.8 13   5 M (0.384615385 0.615384615) *
   3) concave_points_mean>=0.04923 150  15 M (0.100000000 0.900000000)
     6) area_worst< 785.8 26  12 B (0.538461538 0.461538462)
      12) texture_worst< 25.985 12   0 B (1.000000000 0.000000000) *
      13) texture_worst>=25.985 14   2 M (0.142857143 0.857142857) *
     7) area_worst>=785.8 124   1 M (0.008064516 0.991935484) *
```

On peut voir sur les feuilles l'arbre de décision, les nombres de classes mal classifiés est très faibles entre 0 et 5 (5,5,0,2,1).

Le graphe ci-dessous montre les variables par degré d'importance permettant la séparation entre les classes.

Ainsi les top 3 variables importantes permettant la séparation entre les classes sont ''perimeter_worst'', ''area_worst'' et ''radius_worst''.

Ce qui veut dire qu'on peut faire une première reconnaissance entre les deux classes à travers ces variables.



5. Conclusion :

En conclusion, ‘’Decision Trees’’ dans notre cas ont eu une bonne performance prédictive. Dans le cas du cancer du sein, le plus tôt est détecté le mieux, ainsi dans la matrice de confusion prédire une masse mammaire maligne comme étant bénigne peut être très dangereux et vraiment à éviter. Ainsi on peut faire générer plus d’instance maligne pour mettre plus d’équilibre dans les données sinon procéder à l’over-sampling. Une autre alternative sera tout simplement de tester d’autres ‘’classifiers’’ pour une meilleure performance prédictive comme le cas du ‘’Random Forest’’

Les masses de cellules mammaires maligne représentent des caractéristiques relativement très supérieur à ceux bénignes. Voici ces variables qui définissent cette différence par degré d’importance selon les ‘’Random Forest’’