# Machine Learning Project

**Prediction of relapse after 5 years in Cancer from micro-arrays data**

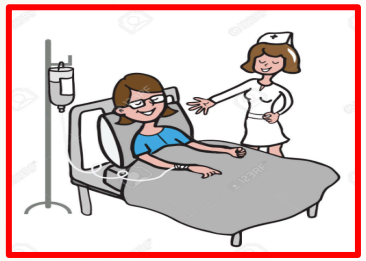


Master 2 bioinformatique : parcours analyse de données biologique

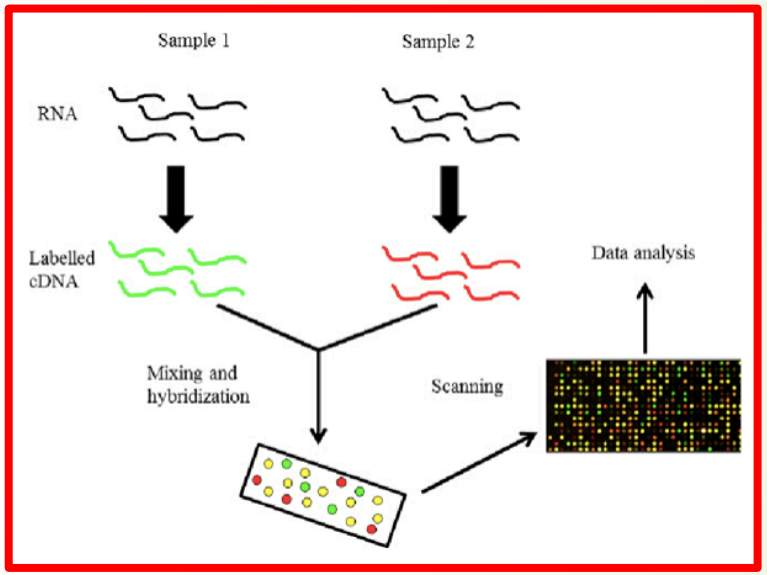
**Etudiant** :Hans CERIL
**Enseignante** : Valérie Monbet

# Contexte biologique

Patient atteint du cancer du sein au stade précoce



184 échantillons




Les chercheurs ont recueillis des niveaux d'expréssion génique pour 4654 gènes.

***xtrain.txt***

Ablation chirugicale de la tumeur

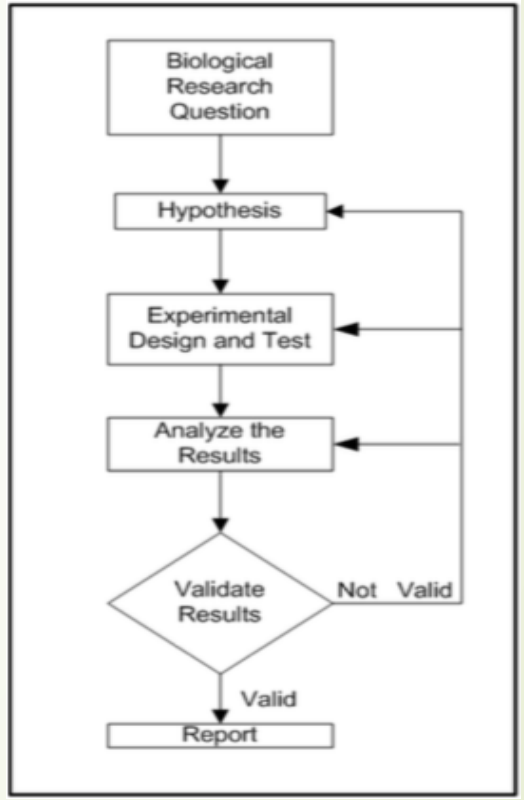
Rechute ➔ +1
Autres ➔ -1

***ytrain.txt***

# Approche Machine Learning

**Pourquoi l'utilisation du machine Learning dans ce contexte biologique ?**

- Le diagnostic et le pronostic d'un type de cancer sont devenus une nécessité dans la recherche sur le cancer.
- La tâche la plus commune dans le processus d'apprentissage est la classification.
- Modèle de classification ➔ 2 types d'erreurs :
  - ✓ Erreurs de classification sur les données d'apprentissage
  - ✓ Erreurs attendues sur les données test
- ✓ Un bon modèle de classification doit bien s'adapter à l'ensemble d'apprentissage et classifier avec précision toutes les instances
- Capacité des outils ML à détecter les principales caractéristique des ensemble de données complexes révèle leur importance.



# Data Set

- Image d'expréssion des données géniques : xtrain.txt
- Les densités de couleurs et les valeurs d'expréssion correspondantes sont affichées sur la barre de couleur verticale droite de la figure
- Les échantillons sont montrés sur l'axes X (184 échantillons) tandis que les gènes sont montrés sur l'axe des Y (4684 gènes)

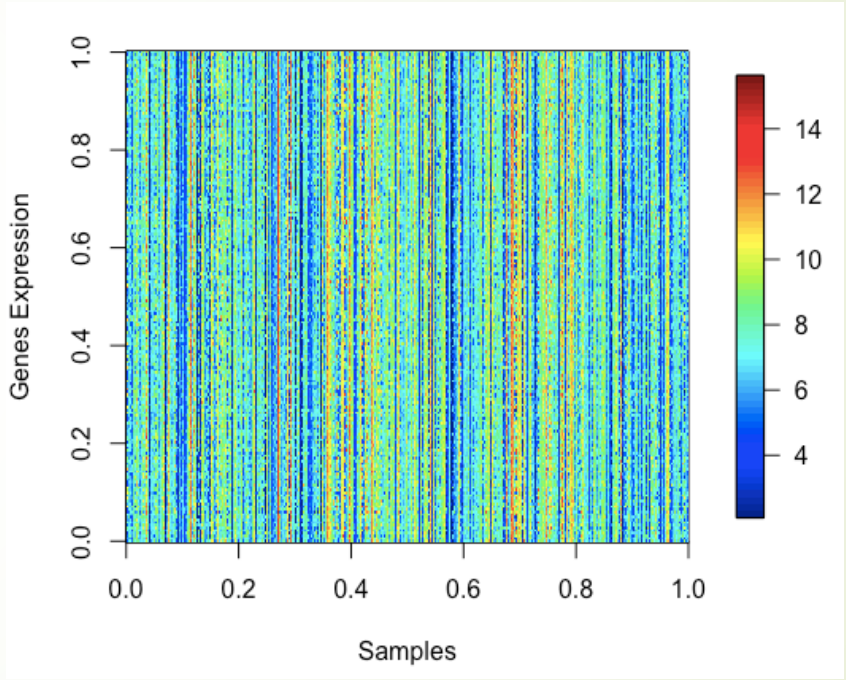
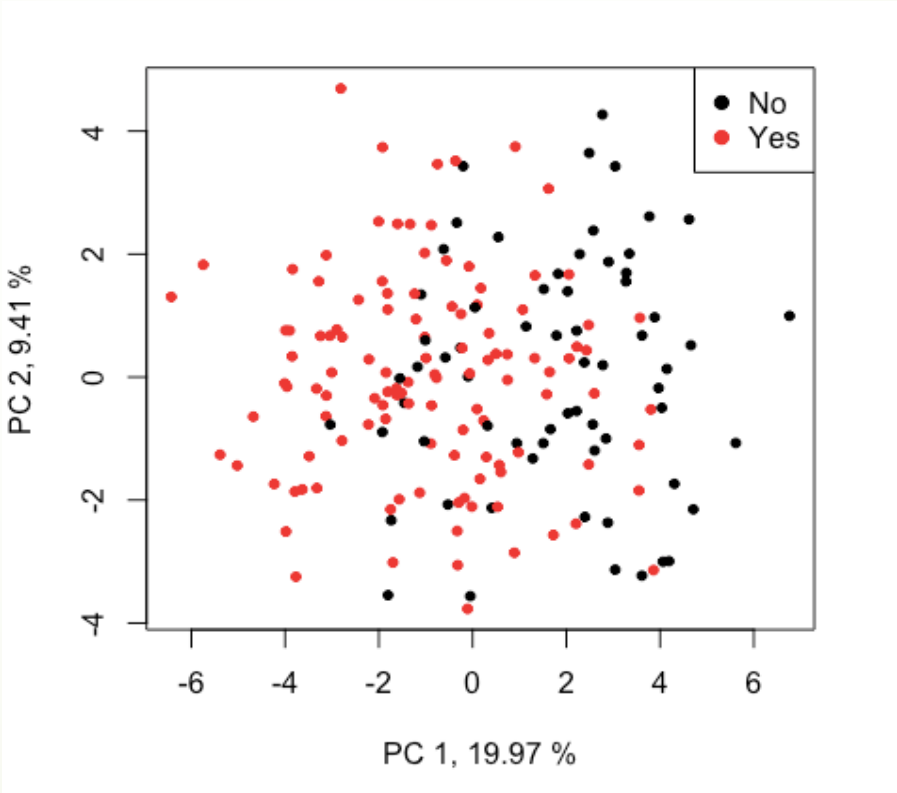
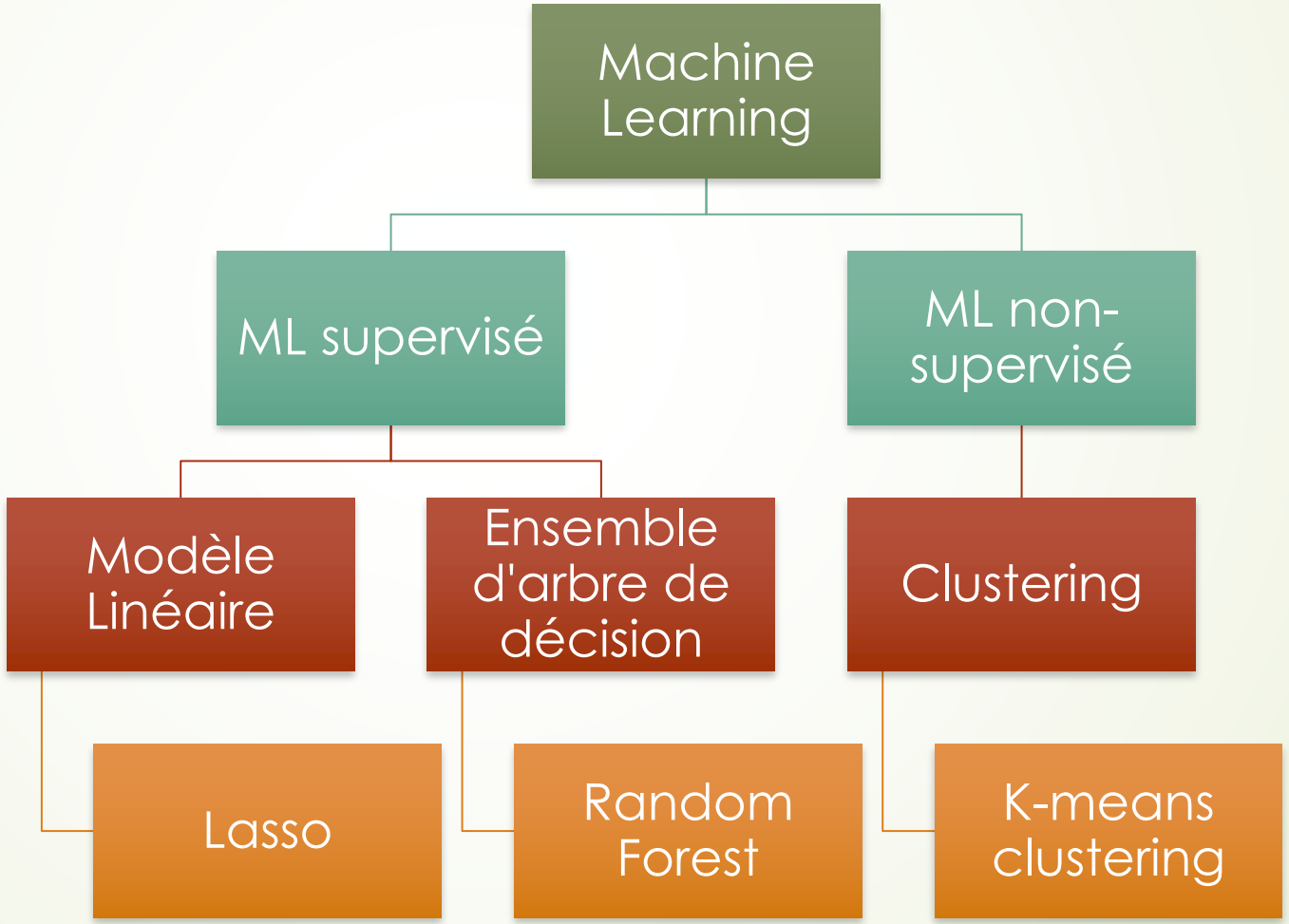



***Image plot of expression values***

# Observation et Vérification des données

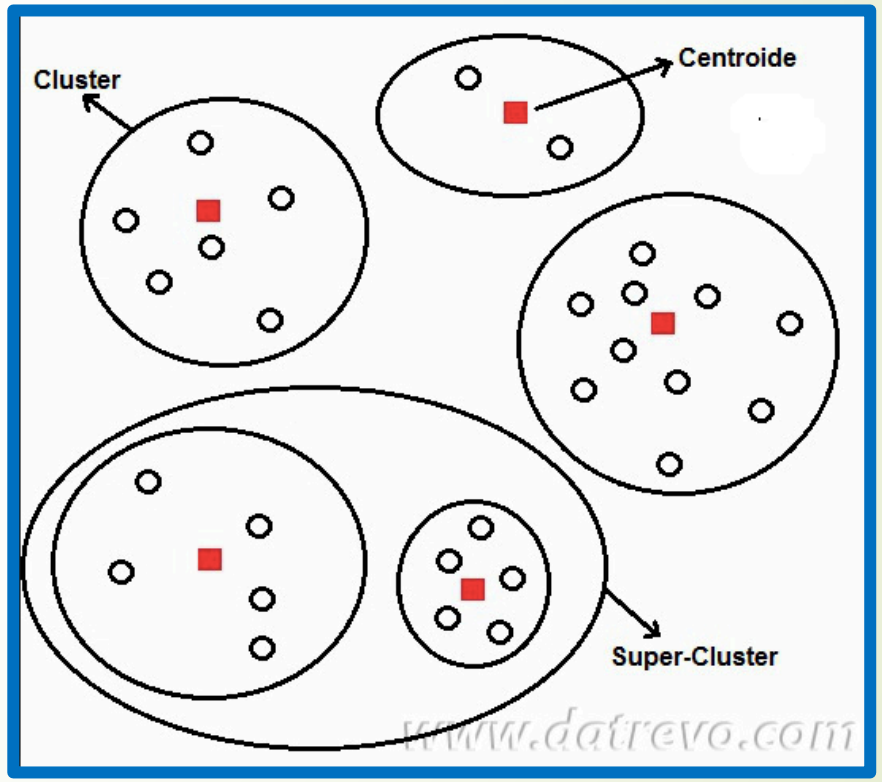


# Méthodes Machine Learning



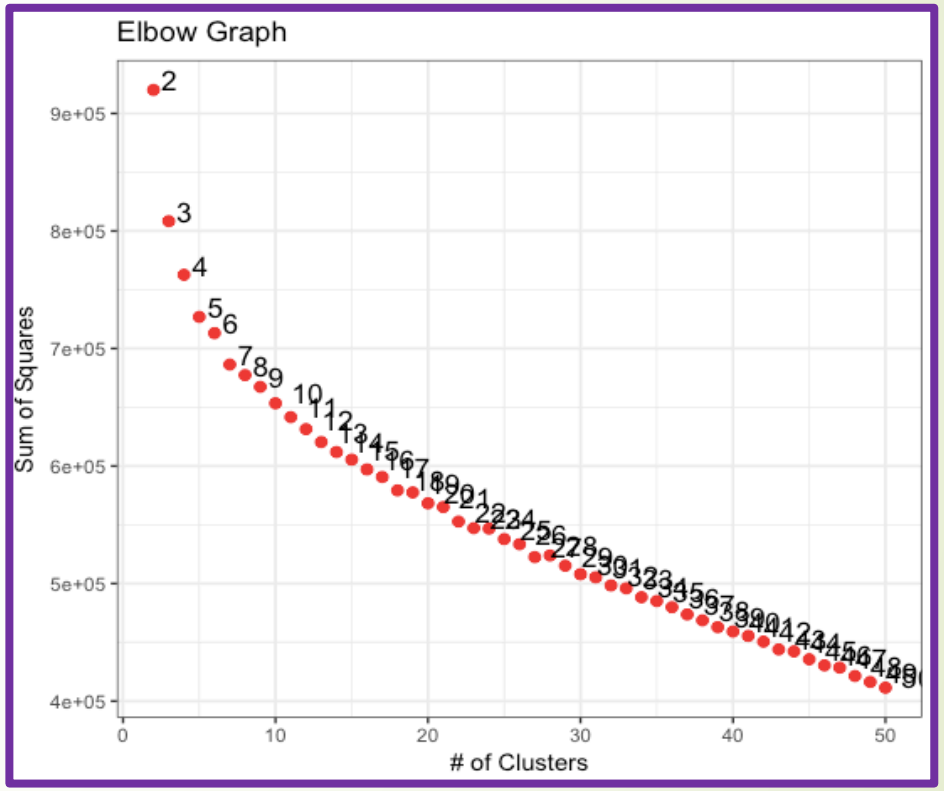
# K-means

- K-means est un type d'apprentissage non supervisé
- Le but de cet algorithme est de trouver des groupes dans les données
- Nombre de groupes représentés par la variable K
- L'algorithme fonctionne itérativement pour affecter chaque point de données à l'un des K groupes en fonction des caractéristiques fournies.
- Les points de données sont regroupés en fonction de la similarité des entités.

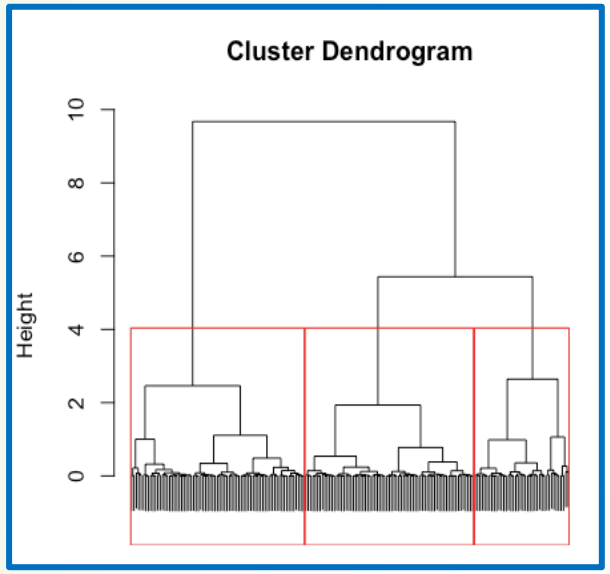


# K-means

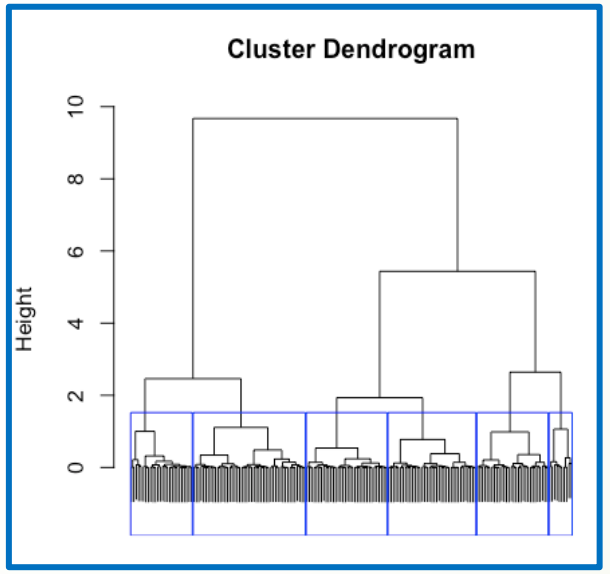
- Méthode Elbow : méthode d'interprétation et de validation de la cohérence au sein de l'analyse de cluster.
- Conçue pour aider à trouver le nombre approprié de clusters dans un ensemble de données.
- Cette méthode considère le pourcentage de variance expliqué en fonction du nombre de clusters: il faut choisir un nombre de clusters de sorte que l'ajout d'un autre cluster ne donne pas une bien meilleure modélisation des données
- Le « coude » ➔ 4 ?



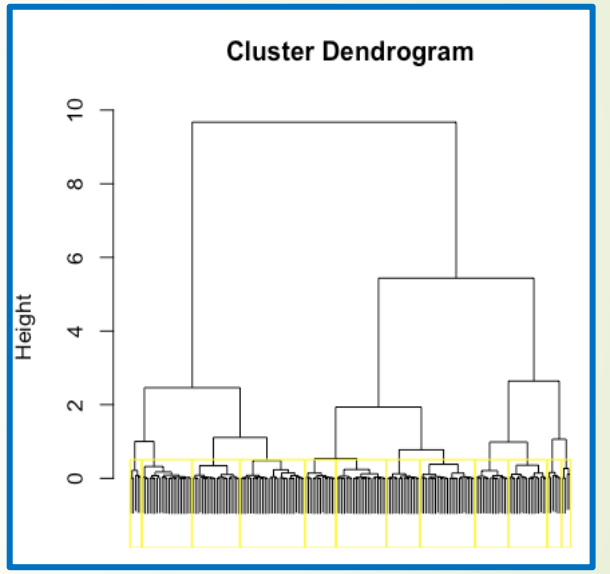
# K-means




K = 3

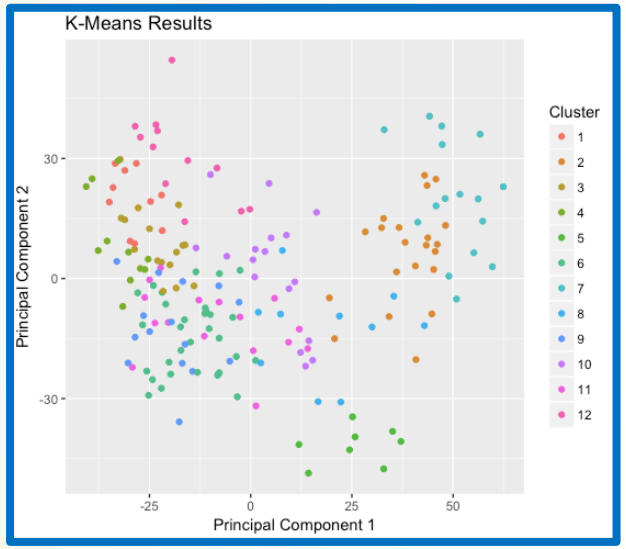


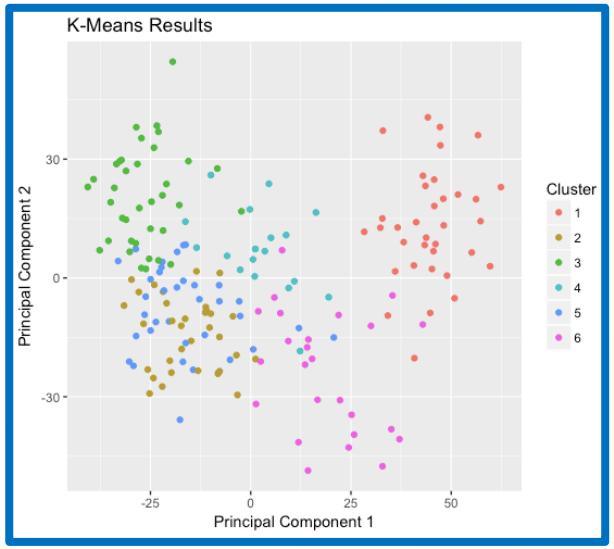

K = 6

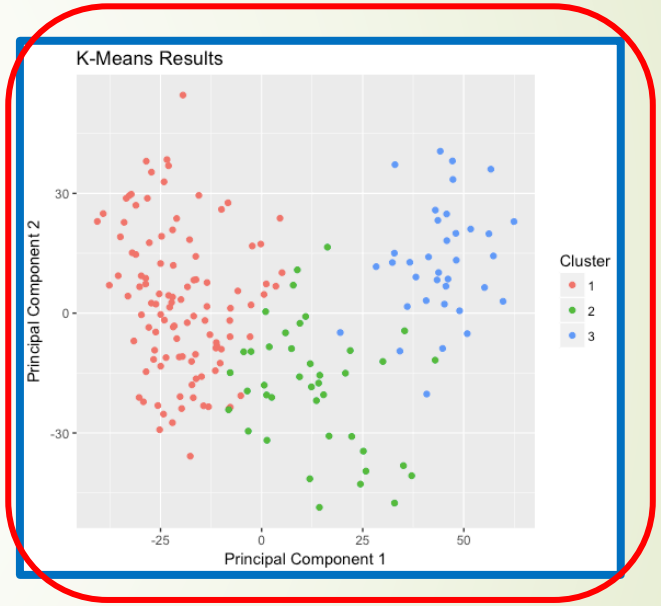



K = 12

# K-means



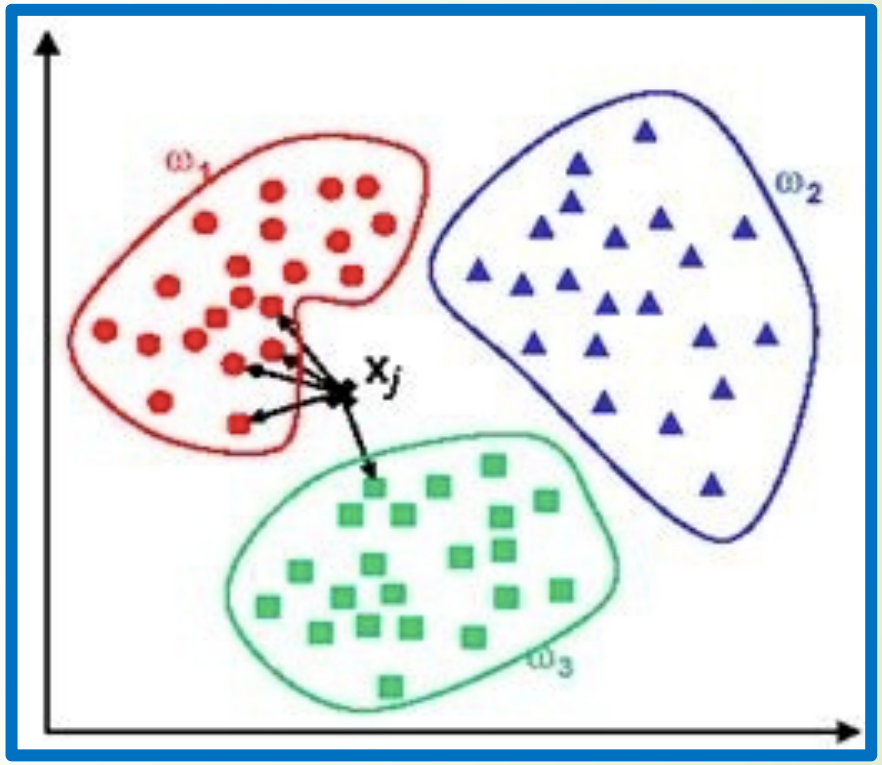




***Meilleur cluster***

# KNN

- Algorithme simple qui stocke tous les cas disponibles et classifie les nouveaux cas par un vote majoritaire de ses voisins k. Ces algorithmes séparent les points de données non marqués en groupes bien définis.
- Forme un vote majoritaire entre les K instances les plus similaires à une observation «invisible» donnée.
- La similarité est définie selon une métrique de distance entre deux points de données

$$d(x, x') = \sqrt{(x_1 - x'_1)^2 + (x_2 - x'_2)^2 + \ldots + (x_n - x'_n)^2}$$



# KNN

- k (nombre de voisins) ➔ L'hyper-paramètre que l'on va chercher à optimiser pour minimiser l'erreur sur les données test.
- Pour trouver le K optimal, on va simplement tester le modèle pour k5, K7, K9.
- La **courbe ROC** permet de visualiser comment la spécificité et la sensibilité d'un modèle évolue en fonction de ce seuil.
- KNN plus performant ➔ KNN7 -> classifie au mieux les données et dans ce cas précis reconnait au mieux les gènes qui sont exprimé différemment en fonction des labels..

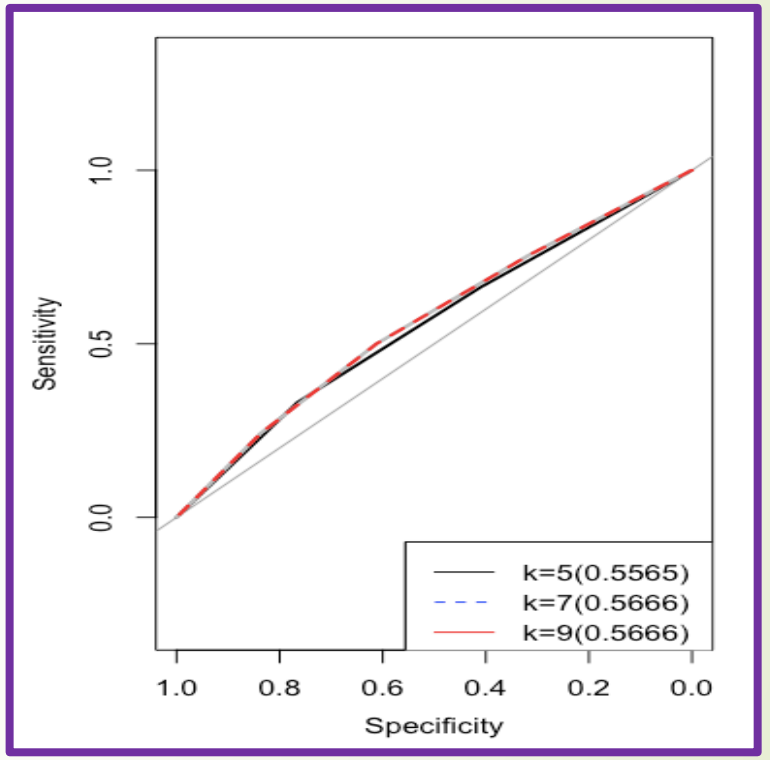
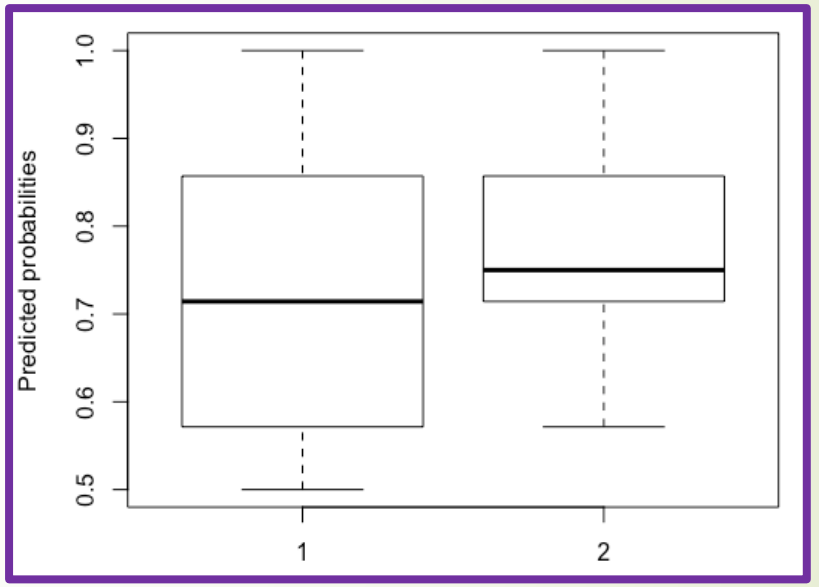



*Figure : An indicative ROC curve of two classifiers: (a) Random Guess classifier (red curve) and (b) A classifier providing more robust predictions (blue dotted curve).*

# KNN

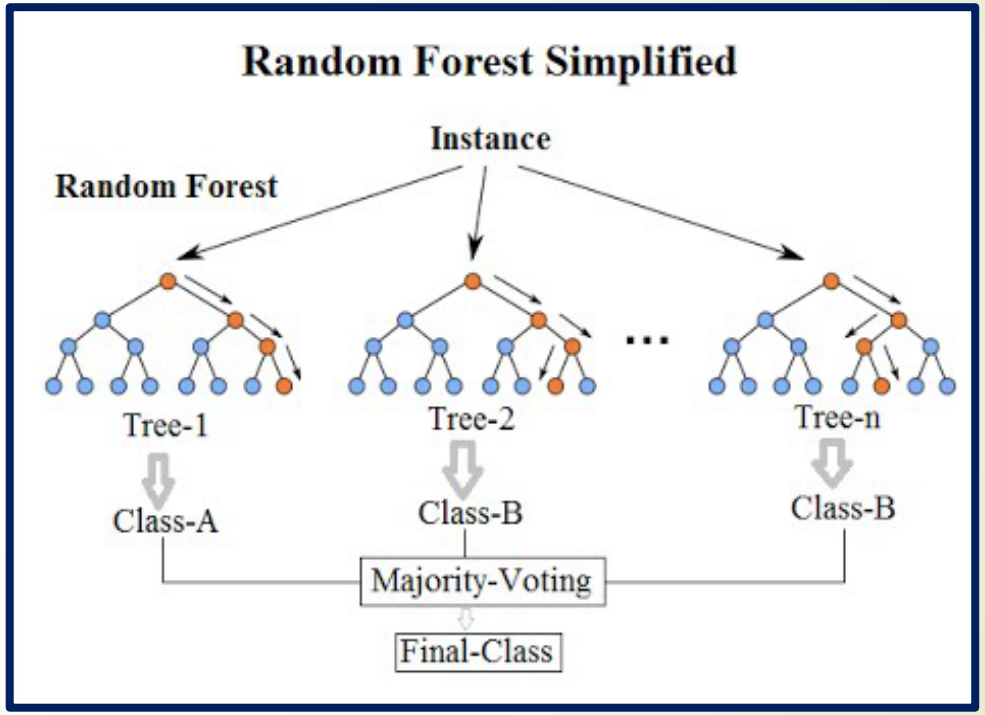
- Différence de répartition de l'expréssion des gènes dans les labels 1 par rapport au label 2
- La figure 3 montre les probabilités prédites du modèle par rapport au label.
- La médiane de la probabilité prédite du label 1 est de 72%  et de 75% pour les label 2.
- Les probabilités prédites les plus faibles
- Meilleurs distribution des labels 1.




*Boxplot de probabilités prédites à partir du modèle KNN pour k=7 Les probabilités prédites.*

# Random Forest

- Les forêts aléatoires ➔ collection d'arbres de décision ou chaque arbre est légèrement différents des autres.
- Injection aléatoires dans la construction d'arbres pour s'assurer que chaque arbre est différent.
- Décorréler les différents arbres qui sont générés sur les différents échantillons « bootstrap » à partir des donner d'entraînement.
- Calcul la moyenne des arbres, nous pouvons réduire la variance et à améliorer la performance des arbres de décision sur l'ensemble de test et éventuellement éviter le sur-apprentissage.



# Random Forest

- 500 sous-ensembles de données sont pris aléatoirement
- Pour chaque sous groupe, un arbre de décision est crée
- Des variables de segmentation sont choisies aléatoirement, et l'arbre est splité en suivant la meilleure segmentation.
- Les 2/3 des données dans « Train » sont utilisées pour le l'apprentissage et les 1/3 sont utilisées pour valider les arbres.
- Le nombre de variables testé aléatoirement lors du split de chaque nœud ➔ 6.
- « Out Of Bag estimate of error rate » ➔ 28.82 %

```
svm = randomForest(Y~.,data = TRAIN,probability=TRUE)
```

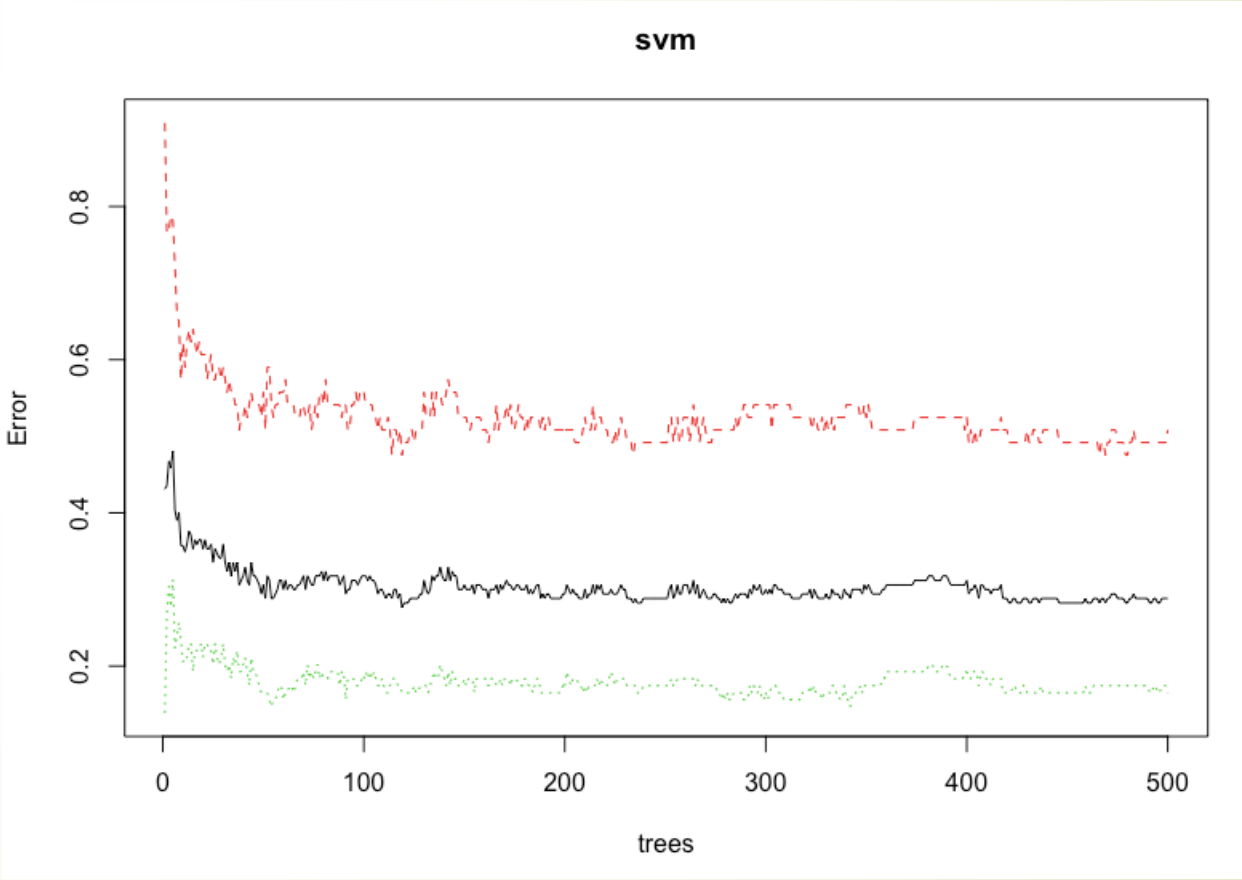
```
> print(svm)

Call:
 randomForest(formula = Y ~ ., data = TRAIN, probability = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 6

        OOB estimate of  error rate: 28.82%
Confusion matrix:
   1  2 class.error
1 30 31   0.5081967
2 18 91   0.1651376
```

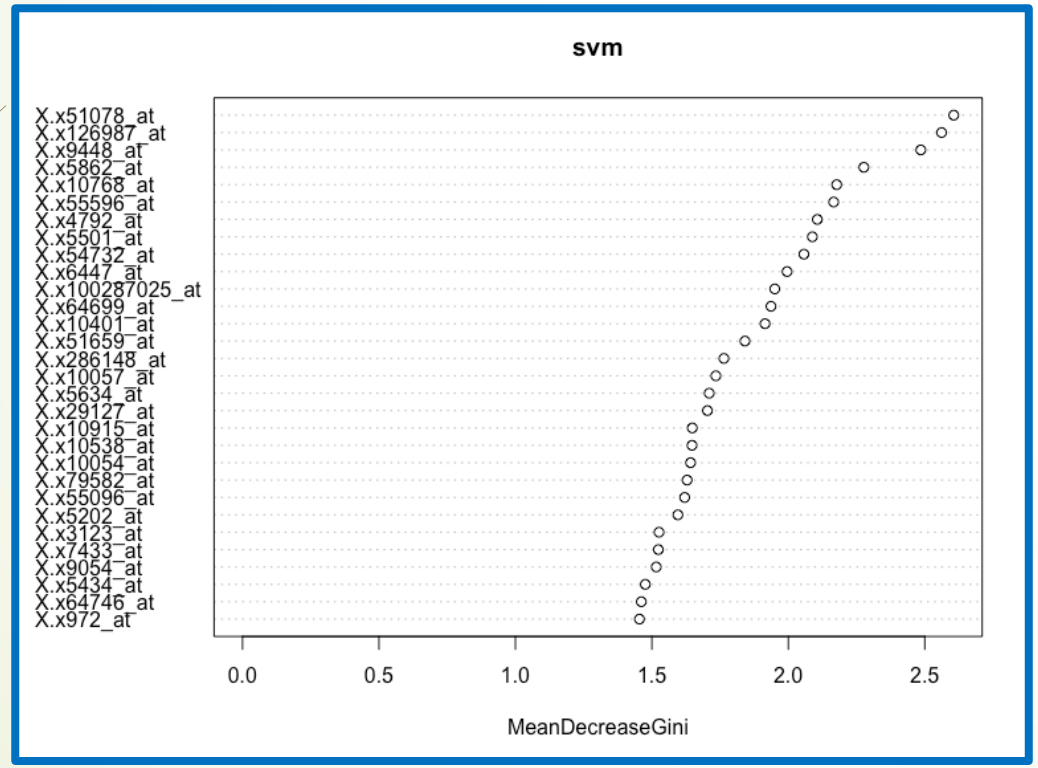
# Random Forest

- Quand le nombre d'arbre augmente ➔ Nombre d'erreur diminue et devient ensuite de plus en plus constant

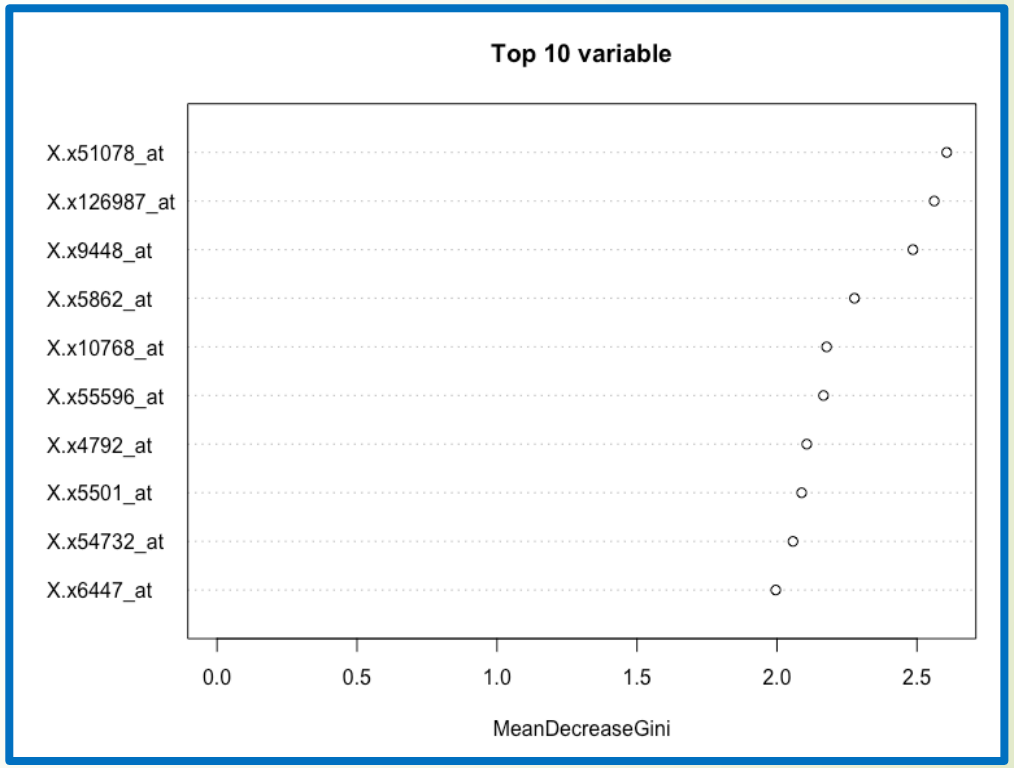


# Random Forest

**> varImpPlot(svm)**




***> varImpPlot(svm, sort = T, n.var = 10, main = "Top 10 variable")***



# Random Forest

- ***Avantages :***

  - Bonnes performances en prédiction
  - Paramétrage simple ( **B** et **m** )
  - Pas de problème d'overfitting ( on peut augmenter B)
  - Mesure de l'importance des variables
  - Evaluation de l'erreur intégrée (OOB)
  - Possibilité de programmation parallèle

- ***Inconvénients :***

  - Problème si nombre de variables pertinentes ➔ les arbres individuels risquent de ne pas être performants.

# Conclusion

- Application dans la prédiction et le pronostic du cancer.
- La plupart des études qui ont été proposées se concentrent sur le développement de modèles prédictifs utilisant des méthodes de ML supervisées et des algorithmes de classification visant à prédire des résultats valides de la maladie.
- Sur la base de l'analyse de leurs résultats, il est évident que l'intégration de données hétérogènes multidimensionnelles, combinée à l'application de différentes techniques de sélection et de classification des caractéristiques, peut fournir des outils prometteurs pour l'inférence dans le domaine du cancer.